Universidad Complutense de Madrid

Grado en Ingeniería Informática

Trabajo de Fin de Grado

Comparative study of the effectiveness of existing methods for low-rate DDoS attacks detection

Marta Pastor Puente

Directed by
Juan Antonio Clemente Barreira
Juan Carlos Fabero Jiménez

Junio, 2019

Abstract

Denial-of-Services (DoS) attacks are nowadays one of the main problems for small and large companies as they entail a high recovery cost in relation to the frequency that they are suffered. Depending on the intensity of the attack launched, these can be defined as high-rate attacks, which seek for a huge shipment of packets in a short space of time, and low-rate attacks, which seek for a continuous delivery of lower proportion of packets for longer time. Being able to detect the latter type is much more complicated due to its similarity with legitimate traffic and, therefore, easily avoids state-of-the-art detection and mitigation measures. The real-time detection of these attacks is certainly a challenge for computer security. This work focuses on presenting some existing detection methods for DoS low-rate attacks as well as analyzing their effectiveness in a simulated traffic environment.

Keywords: network security, Denial-of-Services, attack detection, low-rate, information theory, entropy, Shannon, expectation of packet size, network traffic analysis, traffic simulation

Resumen

Los ataques de denegación de servicios (DoS por sus siglas en inglés) son hoy en día uno de los principales problemas para las pequeñas y grandes empresas, ya que implican un alto coste de recuperación en relación a la frecuencia con que se sufren. La detección en tiempo real de estos ataques es ciertamente un desafío para la seguridad informática. Dependiendo de la intensidad del ataque lanzado, éstos pueden definirse como ataques high-rate, que buscan un gran envío de paquetes en un corto espacio de tiempo, y ataques low-rate, que buscan una entrega continua de menor proporción de paquetes por más tiempo. Ser capaz de detectar ataques low-rate es mucho más complicado debido a su similitud con el tráfico legítimo y, por lo tanto, estos ataques eluden fácilmente las medidas de detección y mitigación actuales. Este trabajo se centra en presentar algunos de los métodos de detección existentes para ataques DoS low-rate, así como en analizar y comparar su efectividad en un entorno de tráfico simulado.

Palabras clave: seguridad de redes, denegación de servicios, detección de ataques, low-rate, teoría de la información, entropía, Shannon, tamaño de paquete esperado, análisis del tráfico de redes, simulación de tráfico

Contents

Acknowledgements

Some people can't believe in themselves until someone else believes in them first.

I want to sincerely thank my tutors, Juan Antonio Clemente and Juan Carlos Fabero, for not giving up on me when I did not even trust myself.

I would also like to thank Ruben, my partner in shakes and illegal raids into the Botanical Garden, for checking that I did not screw up math.

Finally, I want to thank the people who have accompanied me in every step of this path that has been to find my place in the world of computer science, a world dedicated to men and where every day more and more women demonstrate our worth. Thank you, mom, dad, Patricia and Silvia for trusting me blindly. And thank you, Tito, for never letting me sink.

Chapter 1

Introduction

Summarising in one line, Denial-of-Services (DoS) attacks attempt to disable access to resources at the targeted victim that is connected to a computer network. An attack detection system is responsible for identifying an ongoing intrusion whereas a response mechanism attempts to ease the damage that has been caused by localizing the attackers and reducing the intensity of the incursion [1]. In the last few years, with the network migrating to cloud computing environments, the rate of DoS attacks has grown substantially [2].

The difficulty that underlies when defending a system against this type of attacks is due to the fact that said attacks do not target specific vulnerabilities of the victim machine, but the only fact that the target is connected to the network instead. Nowadays there is even a school of thought that considers DoS attacks perfectly legal since they can serve as a means to protest within the Internet. One way or another, preventing and mitigating such attacks has become an issue of critical importance as there is almost nothing the victim can do to be protected from a DoS-type attack.

Based on the rate of packets per second of a DoS intrusion, they can be classified in high-rate and low-rate attacks. High-rate ones were popular in the 90's, but their efficiency is null nowadays because of the improvements that Internet security has experienced ever since. However, low-rate ones are much more complicated to detect due to their similarity with legitimate traffic and, therefore, they easily avoid current detection and mitigation measures which results in a very negative impact for the victim due to the waste of energy and resources that they experience in their systems and, consequently, the increase in their maintenance cost. The real-time detection of these attacks is certainly a challenge for computer security.

For the aim of our work, we will evaluate and compare existing detection methods for low-rate attacks, contrasting the metrics employed in each one as well as their effectiveness based on the percentage of successful detections tested on large networks traffic captures. Moreover, these methods will be implemented using a network simulator to verify whether the effectiveness of the method is maintained for smaller home networks.

Introducción

Resumiendo en una línea, los ataques de denegación de servicios (DoS) intentan deshabilitar el acceso a los recursos de la víctima atacada que se encuentra conectada a la red. Un sistema de detección de ataques se encarga de identificar una intrusión en curso, mientras que un mecanismo de respuesta intenta aliviar los daños causados por la localización de los atacantes y reducir la intensidad de la incursión [**?**]. En los últimos años, con la migración de las redes a entornos de cloud computing, la tasa de ataques DoS ha crecido sustancialmente [2].

La dificultad que subyace a la hora de defender un sistema contra este tipo de ataques se debe a que dichos ataques no se dirigen a vulnerabilidades específicas del equipo víctima, sino al hecho de que el objetivo se encuentra conectado a la red. Hoy en día existe incluso una escuela de pensamiento que considera que los ataques DoS son perfectamente legales ya que pueden servir como medio de protesta dentro de Internet. De una forma u otra, prevenir y mitigar tales ataques se ha convertido en un tema de vital importancia, ya que no hay casi nada que la víctima pueda hacer para protegerse de un ataque de tipo DoS.

Basándonos en la tasa de paquetes por segundo de una intrusión DoS, se pueden clasificar en ataques high-rate y low-rate. Los ataques high-rate eran populares en los años 90, pero su eficiencia resulta nula hoy en día debido a las mejoras que la seguridad en Internet ha experimentado desde entonces. Sin embargo, los ataques low-rate son mucho más complicados de detectar por su similitud con el tráfico legítimo y, por lo tanto, eluden fácilmente las medidas actuales de detección y mitigación, lo que resulta en un impacto muy negativo para la víctima debido al gasto de energía y recursos que experimenta en sus sistemas y, en consecuencia, al aumento del coste de mantenimiento. La detección en tiempo real de estos ataques es sin duda un reto para la seguridad informática.

Para ello, evaluaremos y compararemos algunos de los métodos de detección para ataques low-rate existentes, contrastando las métricas empleadas en cada uno de ellos, así como su efectividad en función del porcentaje de detecciones exitosas probadas en capturas de tráfico de grandes redes. Además, estos métodos se implementarán utilizando un simulador de red para verificar si la eficacia de dichos métodos se mantiene en redes domésticas de menor tamaño.

Chapter 2

State of the art

2.1 DDoS attacks

Traditionally, DoS attackers target a server, which is providing a service to its consumers. Behaving like a legitimate customer, DoS attackers try to flood it in a manner such that the service becomes unavailable due to a large number of pending requests and overflowing the service queue [3]. A different flavour of DoS is Distributed Denial-of-Service (DDoS), where attackers are a group of machines targeting a particular service. There is a high rise in the number of reported incidents of DDoS, which makes it one of the most important threats amongst many [4].

A DDoS attack can be defined as a cooperative, multi-source and large-scale attempt to prevent the legitimate access and use of network resources or services, typically by draining connectivity, processing capacity and/or memory of the targeted machine or network. That is, a lot of malicious hosts coordinate to flood the victim with an abundance of attack packets, so that the attack takes place simultaneously from multiple points [5]. We know from classical textbooks that security falls into three categories: confidentiality, availability and integrity. It is obvious that DDoS attacks belong to the availability category [6].

This kind of attacks are typically launched from a large number of previously compromised hosts to attain the goal of the attack, due to the fact that they require a significant amount of bandwidth to be successful. An example diagram of how a DoS attack works is shown in Figure 2.1. DoS attacks can be generated in application, transport, network and physical layers of TCP/IP framework using different protocols, such as ICMP, TCP, UDP or HTTP [7].

There are different ways to perpetrate a DoS attack in order to achieve the disruption of active services. Their two major goals are to consume bandwidth and to overwork the server.

One typical approach consists in sending a stream of packets from the attacker to the victim. This stream consumes some key resources, which makes the service unavailable for

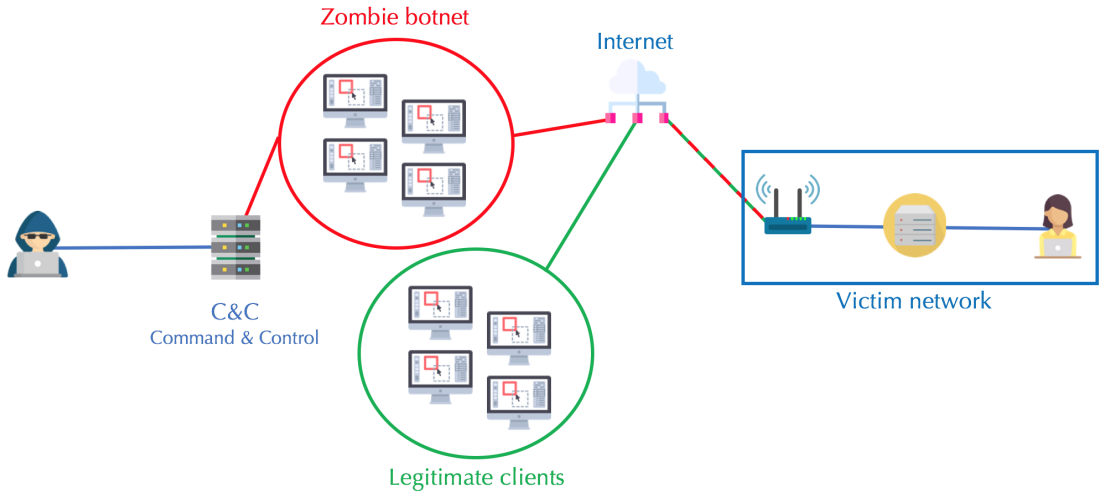


Figure 2.1: Typical DoS attack diagram.

the victim's legitimate clients. Another approach consists in sending some few malformed packets that confuse an application or protocol stack on the victim machine and causes the system to freeze or reboot.

However, why are so many attacks still occurring today and it has not yet been able to stop, or at least mitigate, these attacks?

We have to take into account that the Internet is the largest man-made system in human history. The cyberspace is huge and complex, and it stays in an anarchy status. Moreover, it is impossible to force a security policy to all parties of the Internet, and without collaboration among different ISPs, it is certainly hard to implement said security policy. More importantly, there are even ISPs who support malicious activities for financial or political purposes.

The other main reason for the continuity of attacks to this day is related to the ease of obtaining hacking tools and software in the cyberspace. As a result, an attacker may not need profound knowledge of networking or operating systems to initiate a cyber attack.

2.1.1 How DoS attacks are launched

In general, DDoS attacks can be launched in two forms. The first one aims at crashing a system by sending one or more carefully crafted packets, which are designed based on some vulnerabilities of the victim.

The second form of DoS is to use a large amount of traffic to exhaust the resources of a victim, such as network bandwidth, computing power, operating system data structures,

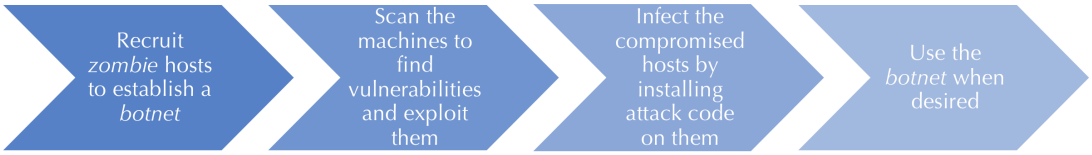


Figure 2.2: Typical flowchart of a DoS attack.

and so on. As a result, the quality of service of the victim is significantly degraded or disabled to its legitimate clients. Compared with the first form, the second form of DoS attack is harder to deal with.

In order to launch an effective DoS attack, cyber attackers have to firstly establish a network of computers known as a *botnet*. In pursuance to organize it, attackers deeply scan the members of the *botnet* in order to find vulnerabilities in the potential hosts that may end up shaping the *botnet* to gain access to them by deeply scanning them. The next step for the attacker is to install malicious code and *malware* programs on the compromised hosts. The hosts running this infected code are known as *bots* or *zombies*. The headquarter of a *botnet* is known as Command and Control (C&C) server [6]. This server is necessary to exist as it will have to communicate with its *bots* for updating the *malware* programs previously installed on them or issuing an attack order and, perhaps, receiving status information from said *bots*.

2.2 History of DoS attacks

The earliest uses of DDoS attacks, often mounted using *botnets* of infected machines, were for extortion and criminal activities. Online gambling industries were the ones who most suffered these threats, since their services were often disrupted at an important time such as on the eve of a major sporting event [8].

It has already been more than 25 years since the first DoS attack took place. Since that first considered as a DoS involuntary attack, performed in 1988 and known as the "Morris worm"[1], a lot of variants have been emerging over the years, although they all share a common result: the disruption of the availability of the target machine or network and its services. It was not until 1996 when an effective DoS attack arose with the goal of said disruption of the system, known as the "Panix attack" [9].

In the late 1990s, the so-called Smurf attack method, an early form of amplification attack, spread quickly and became well known. In a Smurf attack, an ICMP Echo Request packet is sent. The attackers spoof the source address of these packets to be the

[1]The name comes from its creator Robert Tappan Morris, a graduate student at Cornell University.

ultimate attack victim's IP address, and send the traffic to the broadcast address of a network. All of the hosts on the network will do their best to reply to the traffic –sending an ICMP Echo Reply packet in response to every Echo Request packet received–, but in fact they send it to the victim. In this way, a single attacker can noticeably multiply their traffic [9].

DDoS attacks have shown a wide range of different motivations and reasons to be performed over the years, which started as acts of heroism to end up evolving into a mechanism to compete in an invisible war with political and economic purposes. We can see that cyberspace has become a haven for intelligent criminals, who are motivated by significant financial or political reward. All these social reasons have led to use DoS attacks as a tool for profit making, using them to paralyze a competitor's web or to degrade connectivity of users to market rival services, as examples.

In business terms and according to the cyber crime report the Ponemon Institute publishes annually [10], larger organizations experience a higher proportion of costs relating to denial-of-services, malicious insiders and malicious code. DoS attacks have, on average, the second most expensive cybercrime cost weighted by attack frequency, right after malicious insiders whose impact is a 25% greater than the DoS one. Moreover, the cost of these cybercrime activities as a whole affects different industry sectors in a very variable scale as seen in Figure 2.3.

From among the many attackers who target the Internet infrastructure, such as the root DNS servers, about two per year are able to build a *botnet* large enough to cause a noticeable impact. A small subset of DoS attacks is financially motivated. In these, attackers threaten or demonstrate that they can disrupt an e-commerce site and demand a ransom from the victim to prevent further attacks [11]. Another subset of DDoS attacks appear to be politically motivated. These attacks gained wide public attention and they are now closely associated with hacktivism and other political uses.

2.3 Current classification of DoS attacks

Due to the wide variety of cyber-attacks taking place on the Internet today, and their constant evolution towards new scenarios, it is difficult to classify DoS attacks univocally and universally. However, a series of classifications developed by organizations dedicated exclusively to the field of cybersecurity have already been established and can serve as a basis and reference for later developing a classification based on our approach to the work we want to develop.

Among these organizations, we will use the classifications proposed by ENISA and INCIBE. The latter, in particular, highlights two classifications of DoS attacks according to the damage it causes to the victim and the level of the OSI layer that is exploited.

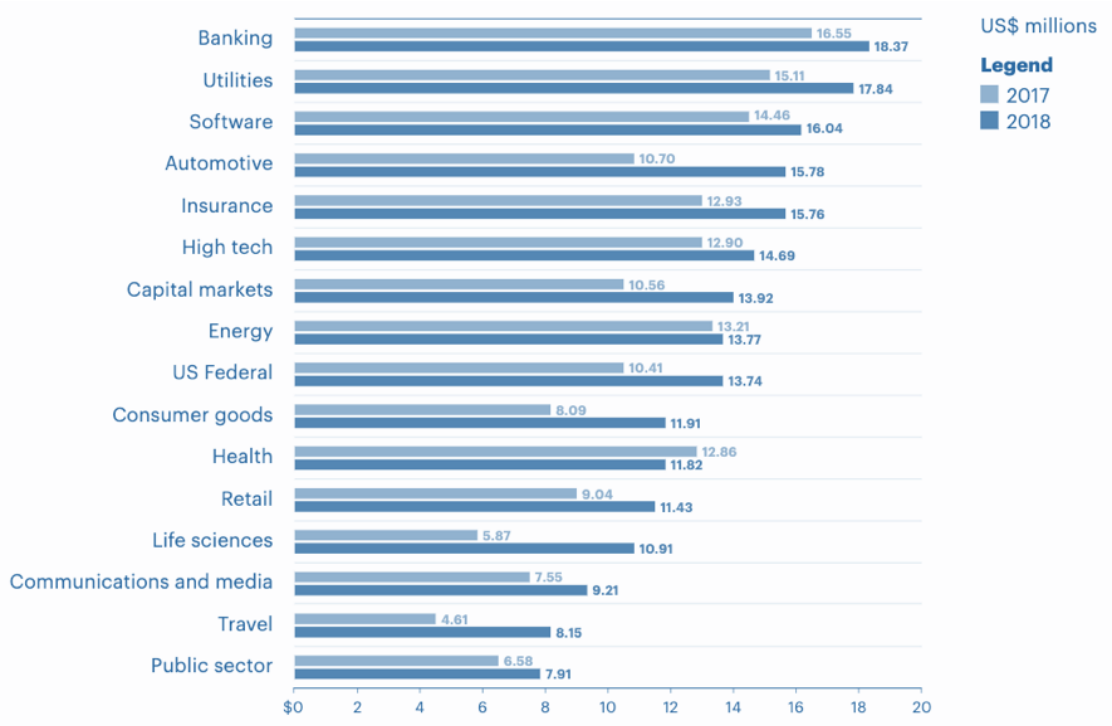


Figure 2.3: Chart of average annual cost of cyberattacks by industry sectors in 2017 and 2018 according to the Ponemon Institute anual report [10].

2.3.1 Based on the kind of damage produced

1. **Saturation:** the main objective of this type of attack is to overload or saturate certain system resources that are key to its proper functioning. These resources may include bandwidth, processor usage, memory consumption, available access to external resources, storage disk space or system power. This attack strategy has been mainly used in recent years by the hacktivist group Anonymous, carrying out a series of attacks that relied on a wide network of attackers.

2. **Modification of configurations:** The main objective of this type of attack is to alter or eliminate the configuration parameters of certain resources that are key to the proper performance of affected systems, being these mainly servers or routers.

3. **Destruction:** The main objective of this type of attack is the modification or destruction of physical components of the affected system. Although initially, in order to carry out this type of attack strategy it was necessary to have physical access to the facilities where the victim systems were hosted, the fact that certain industrial control systems are currently connected to the Internet has made it easier for attackers to carry out this type of attacks remotely.

4. **Disruption:** The main objective of this type of attack consists in the sudden interruption of communications between two or more devices through an alteration of the state of the information that is in transit, so that the transfer of such information is unfeasible. A clear example of this attack strategy would be the illegitimate restart of active Transmission Control Protocol (TCP) sessions.

5. **Obstruction:** The main objective of this type of attack is to obstruct communications between two or more interlocutors, thus preventing both parties from successfully and fully contacting each other. A clear example of this type of attack could be the selective filtering of IP addresses by ISPs based on their reputation level.

2.3.2 Based on the level of OSI layer targeted

1. **Infrastructure level:** In this group we are going to include all the different types of attacks that focus on exploiting vulnerabilities of the systems at network and transport layer level following the OSI layer model. The network protocols that are most often under attack are TCP, UCP and IMCP, as they are the most commonly used protocols for the exchange of information on the Internet.

   However, there are other protocols in these layers that, although they are not used directly as a basis for communication on the Internet, have vulnerabilities that may be critical due to the environment in which they are used, such as the DNP3 (Distributed Network Protocol version 3), widely implemented in industrial environments.

Figure 2.4 shows a classification developed by the North American DDoS protection company, RioRey, which presents a taxonomy of the different types of attacks that can be depicted according to the type of protocol under attack.

2. **Application level:** This group includes all those attacks directed against the last layer of the OSI model, which seek to exploit vulnerabilities inherent to the implemented applications that result in an unavailability of the service for legitimate users. DDoS attacks targeting the HTTP protocol are very common, although effective attacks against SMTP and DNS protocols have also been developed.

2.4 Detection of DoS attacks

To defend against DoS attacks, several countermeasures have already been developed. Anyhow, all of them usually consist of three ordered steps: detection, mitigation and IP trace-back.

In pursuance of achieving attack detection, an Intrusion Detection System (IDS) must be developed. It is defined as some sort of software or hardware used to detect unauthorized traffic or activities that are against the allowed policy of a given network [12]. A simple classification of IDSs is shown in Figure 2.5. They can be classified **based on the audit source location** as either host-based, network-based or a combination of both. IDSs can also be categorized **based on the detection method employed** as one of two types: signature-based or anomaly-based detection [13]. The first one works based on an already existing database with known attack signatures that match certain patterns or strings with a pattern of incoming packets that have been analyzed. The second one consists in an anomaly-based detection method which compares the previous network behavior profile prepared from normal traffic and based on a stored historical with the incoming network behavior at real-time [14, 15].

Moreover, when talking specifically about network-based IDSs, we can also differentiate between various systems of detection **depending on the analysis of the network traffic** we perform – statistics-based, knowledge-based or machine learning-based. In addition, **depending on the granularity of analysis** in the network, we can distinguish among packet-level and flow-level analysis. We will focus in this second type in our work.

Chapter 4 explains two specific methods for low-rate detection that have also been tested for high-rate detection. These attacks fall into the category of anomaly-based methods, as they compare the properties of the traffic under study with the properties of traffic previously characterized as legitimate. In this sense, **information theory-based metrics** and **expectation of packet size** are defined. Both methods calculate, from mathematical models, a series of traffic properties that have resulted to effectively detect changes in network behavior for which traditional detection methods have not proven to be practical. Both methods will be explained in detail further on.

		Attack Types	Attack Matrix Dimensions									
			Nature of IP	Handshake	Source IP Range	Packet Rate	Packet Size	Packet Content	Fragmenting	Session Rate	Session Duration	VERB Rate
TCP BASED	1	SYN Flood	Spoofed	None	Large	High	Small	---	---	---	---	---
	2	SYN-ACK Flood	Spoofed	None	Large	High	---	---	---	---	---	---
	3	ACK & PUSH ACK Flood	Spoofed	None	Large	High	---	---	---	---	---	---
	4	Fragmented ACK	Spoofed	None	Large	Moderate	Large	---	High	---	---	---
	5	RST or FIN Flood	Spoofed	None	Large	High	---	---	---	---	---	---
	6	Synonymous IP	Spoofed	None	Single IP	High	---	---	---	---	---	---
	7	Fake Session	Spoofed	None	Large	Low	---	---	---	---	---	---
	8	Session Attack	Non-Spoofed	Yes	Small	Low	---	---	---	Low	Long	---
	9	Misused Application	Non-Spoofed	Yes	Small	Variable	---	---	---	High	Short	---
TCP HTTP BASED	10	HTTP Fragmentation	Non-Spoofed	Yes	Small	Very Low	Small	Valid	High	Very Low	Very Long	Very Low
	11	Excessive VERB	Non-Spoofed	Yes	Small	High	---	Valid	---	High	Short	High
	12	Excessive VERB Single Session	Non-Spoofed	Yes	Small	Low	---	Valid	---	Low	Moderate	High
	13	Multiple VERB Single Request	Non-Spoofed	Yes	Small	Very Low	Large	Valid	---	Low	Long	High
	14	Recursive GET	Non-Spoofed	Yes	Small	Low	---	Valid	---	Low	Short	Low
	15	Random Recursive GET	Non-Spoofed	Yes	Small	Low	---	Valid	---	Low	Short	Low
	16	Faulty Application	Non-Spoofed	Yes	Small	Low	---	Valid	---	Low	Short	Low
UDP BASED	17	UDP Flood	Spoofed	---	Very Large	Very High	Small	Not Valid	---	---	---	---
	18	Fragmentation	Spoofed	---	Moderate	Very High	Large	Not Valid	High	---	---	---
	19	DNS Flood	Spoofed	---	Very Large	Very High	Small	Valid	---	---	---	---
	20	VoIP Flood	Spoofed	---	Very Large	Very High	Small	Valid	---	---	---	---
	21	Media Data Flood	Spoofed	---	Very Large	Very High	Moderate	Valid	---	---	---	---
	22	Non-Spoofed UDP Flood	Non-Spoofed	---	Small	Very High	---	Valid	---	---	---	---
ICMP BASED	23	ICMP Flood	Spoofed	---	Very Large	Very High	Variable	Not Valid	---	---	---	---
	24	Fragmentation	Spoofed	---	Moderate	Very High	Large	Not Valid	High	---	---	---
	25	Ping Flood	Spoofed	---	Very Large	Very High	Small	Valid	---	---	---	---

Figure 2.4: DDoS taxonomy proposed by RioRey company.

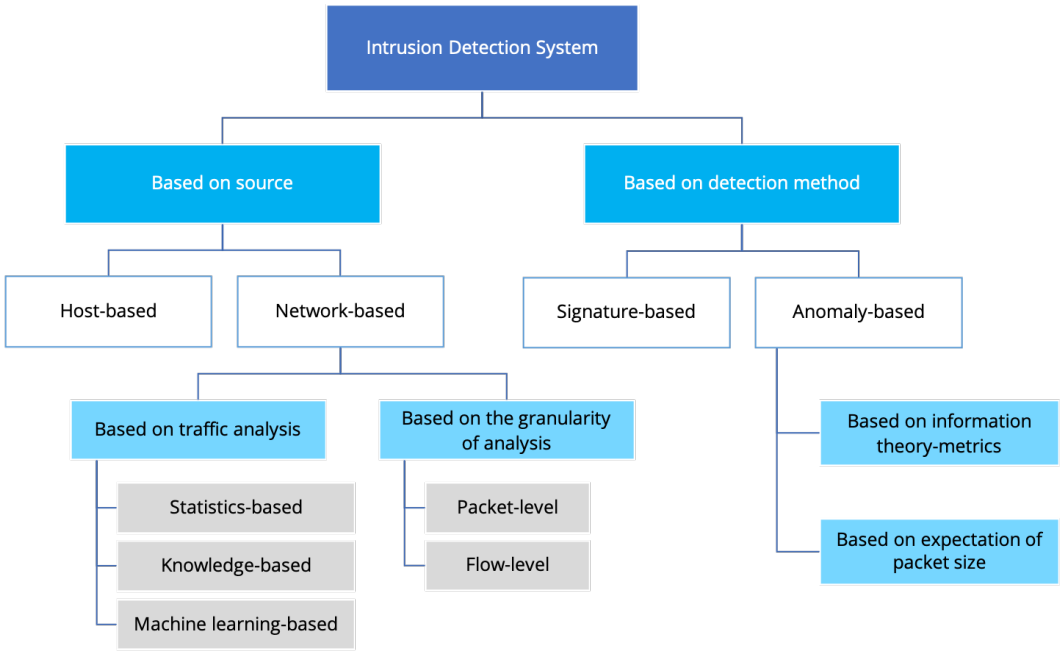


Figure 2.5: Proposed IDS classification.

It is clear that, for detecting low-rate attacks based on the capacity of each network and the behavior of its traffic considered as legitimate, the network-based, machine learning-based approach should be the one adopted. For this purpose, it is interesting to differentiate between an attack and an anomaly. An attack can be defined as a sequence of operations that puts the security of a system at risk, whereas an anomaly is just an event that is suspicious from the the point of view of the security [16].

In a paper published by a group of researchers from the University of Granada [16], anomaly-based intrusion detection systems are described to operate in three phases:

1. Parameterization: the parameters of the system are defined
2. Training: the model of the normal behavior of the traffic is built
3. Detection: the traffic behavior is compared against that of the training phase. If the comparison exceeds a threshold value, a detection alarm is triggered.

Although the characterization of legitimate traffic that occurs during the parametrization and training phases seems to be static and performed only once at the start of the detection procedure, these two steps can be dynamically calculated as the traffic during peak hours is significantly more dense than at dawn.

The majority of current DoS detection methods belong to the signature-based ones and therefore they are incapable of detecting not well-known or new attacks. The reason is that these methods are based on specific attack features. Thus, innovative anomaly-based approaches have been recently proposed to analyze different parameters of attack flows. In order to help detecting these not-known attacks, we will compare existing detection methods based on information theory metrics and expectation of flows packet size. Moreover, we will implement a small network topology to perform low-rate attacks and compare the effectiveness of said detection methods.

Depending on the approach used to monitor and observe network traffic, the analysis of said network can focus whether in individual packets of information or flows of those packets. The latter approach is said to be way more scalable than traditional packet-based traffic analysis. As flow monitoring grasps the complete chain of packet observation, data classification and data analysis, we will classify traffic within flows instead of as individual packets in order to better detect mistrustful network behaviour.

For a deeper understanding of the comprehensive comparison of existing low-rate attack detection methods that will be developed in Chapter 4, it is relevant to know other especially network-based detection methods that have proven not to be as effective for the concrete detection of low-rate attacks. Amongst these methods, we will define activity profiling, sequential change-point detection and wavelet analysis.

Activity profiling consists in calculating the average packet rate for a specific network flow. Consecutive packets with similar header fields such as both source and destination addresses, ports and protocol represent a network flow. The average packet rate

can be determined by the time elapsed between two consecutive matching packets. The average packet rates of all incoming and outgoing flows are used to calculate the total network activity by dividing the sum over the average packet rates. Moreover, individual flows with similar characteristics can be grouped in a cluster. The average packet rate of the clusters will be used to detect the attack based on an increase in the average packet rate within the clusters.

On the other hand, **sequential change-point detection** is based on detecting a change in statistical models as soon as possible after the occurrence of said change, thereby reducing the probability of triggering a false alarm. More specifically, the problem of detecting an attack can be formulated and solved as a change-point detection problem by detecting a change in the distribution or model with a minimal average delay controlling, at the same time, the rate of false detections [17]. This detection method filters the traffic at the victim, according to source and destination addresses, ports and protocol. The filtered traffic is treated as a time series. This type of algorithms is applied to DoS attacks by comparing the actual average for the traffic in the time series with the expected average from a time series sample computed in advanced.

Finally, **wavelet[2] analysis** works with spectral components in wavelets as input signal. This method counts with an inherent time-frequency property that allows splitting signals into different components at several frequencies. Wavelet transform can be used to analyze and characterize flow-based traffic behaviors by splitting signals into different components at three ranges of frequencies, for example. Thereby, low frequency components may correspond to patterns over a long period; mid frequency components capture daily variations in the flow data, and high frequency components analyze short term variations. Wavelets provide time and frequency characterization for a signal, as opposite to traditional Fourier analysis, which only provides a frequency characterization [18]. The time-localized anomalous signals can be separated from the noise signals by wavelets and by analyzing each spectral window's energy, the anomalies can be determined. The main advantage of wavelet analysis is that it is able to capture complex temporal correlation across multiple time scales with very low computational complexity [19]. However, this method has its own limitations since low frequency scans have resulted not to be accurate enough as anomalous patterns are not correctly detected.

These traditional methods have a number of limitations when it comes to carrying out effective detection of DoS attacks. Amongst said limitations, the main one is undoubtedly the difficulty of detecting anomalous behaviour based on long-term observation, even if comparing current traffic with an historic of the traffic observed in advance. Thereby, the use of customized low-rate attack detection methods is more interesting as they have combined the advantages of each of these traditional methods with the inherent feature of long-term of the low-rate attacks.

[2]A wavelet is a mathematical function used in digital signal processing and image compression and its principles are very similar to those on Fourier analysis. Wavelets make it possible to recover weak signals from noise.

Chapter 3

Low-rate DoS attacks

3.1 Definition of low-rate DoS attacks

Low-rate attacks are defined as those denial of service attacks that, having as their main objective to affect the availability and performance of the targeted system, present a traffic distribution and a packet sending rate similar to legitimate traffic, so that the difficulty of effective detection is increased in contrast to traditional DoS attacks, known as high-rate.

Traditional detection methods, previously explained, focus on a detection strategy based on the comparison of incoming traffic against a known pattern, specifically monitoring parameters such as the number of packets per second sent or the number of simultaneous requests made from an attacking machine.

Low-rate attack traffic cannot be analyzed by following a known pattern of behavior that could trigger the alarm, as they are indistinguishable from legitimate traffic. Therefore, these methods are not effective and new techniques based on different approaches have been developed in recent years.

3.2 Proposed classification for DoS attacks

Based on the classification framework proposed by Mirkovic and Reiher [20], the most interesting categories have been selected in this work in order to encompass and better explain existing low-rate detection methods later explained, depicted in Figure 3.1.

3.2.1 By source address validity

Making a differentiation based on the validity of the source address may seem unnecessary at first for what concerns the detection of a DoS attack. However, it is of great interest for a later implementation of the mitigation system, since spoofed addresses can be more difficult to block without negatively affecting access to the service to legitimate users.

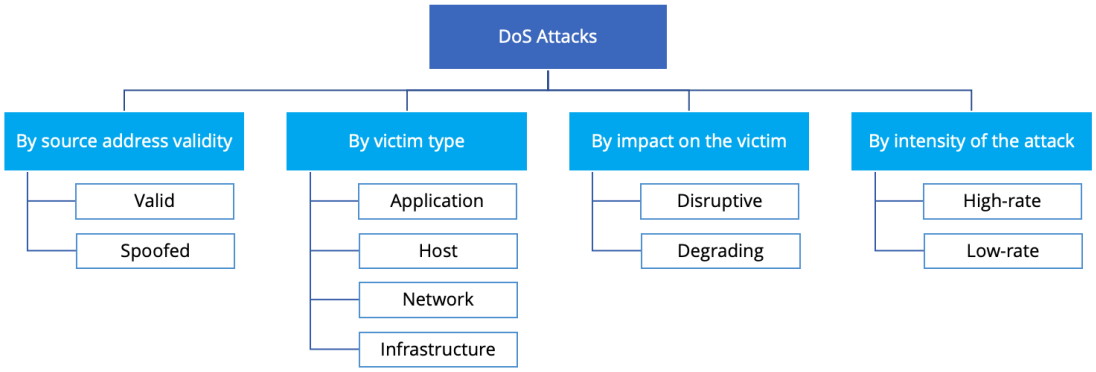


Figure 3.1: Proposed DoS attacks classifying framework based on Mirkovic and Reiher's taxonomy.

Thus, using spoofed source addresses is highly desirable for the attacker in order to avoid accountability. A clean example of this is the Smurf attack, previously explained. Moreover, this kind of attack may spoof source addresses by using a reserved set of them or part of an assigned but not used address space of the targeted network. Nevertheless, current routers and firewalls rules can easily detect these spoofed addresses by comparing if incoming packets have been actually sent from an internal address and if not, they are discarded. Anyhow, it is not a necessary condition to perform a DDoS attack since those forwarding to invalidate certain applications or protocol features must use valid source addresses [20]. In this way, spoofing is certainly more preferable for attackers although a DoS attack can also be effectively performed with valid source addresses.

3.2.2 By victim type

As attacks do not necessarily need to target a single host machine, a differentiation between the type of targeted victim is assuredly required. As indicated in Figure 3.1, this criteria yields four types of attacks: Application, Host, Network and Infrastructure-based attacks.

Application attacks exploit some elements of a specific application on the victim host hence disabling legitimate clients from using said application and probably clogging resources of the attacked host machine. In order to avoid them, each application would have to be securely modeled and its performance monitored to account for possible attacks [20].

Next, the so-called host attacks seek to completely disable the victim machine through an overload or through undermining its communication mechanism with the network. The high packet volume required to accomplish that purpose facilitates their detection.

Network attacks consist in consuming the incoming bandwidth of the target network with packets whose destination address belongs to the target network's address space. This kind of attack can also be easily detected due to their high volume of packets.

Finally, infrastructure attacks focus on bringing down a crucial distributed service for the Internet network –such as power plants or DNS server farms– independently if it is globally or locally located. The main feature of these attacks is not the mechanism deployed to disable the target infrastructure, but the simultaneity of the attack on multiple instances of said critical services in the Internet as a whole.

3.2.3 By impact on the victim

Possibly the most interesting differentiation in this classifying framework is to focus on the impact on the victim; hence the disruptive and degrading attacks. Although disruptive attacks are the kind most currently known attacks belong to, degrading ones are considerably more interesting. Whereas the former ones focus on completely denying the victim's service to its clients, the purpose of the latter is to consume part of the victim's resources. As this variety of attacks does not result in total service disruption, they could remain undetected for a significant amount of time, thereby causing a huge damage on the victim in business terms.

For large companies that offer their services in the network such as Google or Netflix, both types of attacks are equally dangerous as they prevent the correct use of the only service for which their customers pay, which is to have it available on the Internet.

A disruptive attack can be very expensive in temporary terms immediately after suffering the DoS attack since it can take several hours to restart the fallen servers. However, companies suffer much more the effects of degrading attacks in the long term, because by not detecting them so easily and not throwing the network down completely, but simply hindering and reducing the quality of service, users may end up being unsubscribed from service after a while for not meeting the quality expectations they had about said service.

Not all DoS attacks involve flooding the network, such as the classic volumetric disruptive attack. Other forms focus on the application layer by making a lot of requests for large files or by imposingly slowing down connections at the application level. Therefore, due to a considerable number of poorly configured servers on the Internet, disruptive attacks using reflection methods[1] are the prevalent ones nowadays and lean to be the methods favored for massive-scale attacks that exploit poorly configured servers.

[1] In a DoS reflection attack, an attacker spoofs his or her IP address with the victim's one and sends request messages for information to servers or hosts on the network that are well-known for responding to such type of messages. From the servers' perspective, it was the victim who sent the original request. All the data from those servers gathers, congesting the target's Internet connectivity if the attacker has much higher bandwidth than the victim since the victim machine gets lots of unsolicited responses that consume all its network bandwidth.

3.2.4 By intensity of the attack

In an initial approach of the classification of Mirkovic and Heiser (Figure 3.1), we considered, from the definition of degrading attack, that the concept of low-rate attack could be adjusted to this same definition. However, after analyzing the aspects that were taken into account when categorizing an attack as disruptive or degrading, we realized that the impact that a DoS attack may have on the victim depends more on the victim's ability to respond to. In contrast, classification in low-rate or high-rate exclusively depends on the ratio of packets per second that are sent. It may be the case of launching a high-rate attack without necessarily crashing the targeted service because it has a powerful mitigation system and in the same way, a low-rate attack may be able to knock down a smaller service such as a home network.

In this way, we can define high-rate attacks as those that involve a large volume of traffic and that are normally designed to stop a service completely – although, as we have explained before, the success of this action depends on the mitigation measures of the attacked machine. They are relatively easy to detect as their traffic profile deviates significantly from the normal traffic profile of the network [21]. On the other hand, low-rate attacks are much more difficult to detect since they are intended to significantly affect the connectivity to the network or service attacked through a constant sending of packets that are coated among legitimate traffic and they can easily evade the traditional anomaly-based detection systems [21]. Specifically, maliciously chosen low-rate DoS traffic patterns that exploit TCP's retransmission time-out mechanism can strangle TCP flows to a small fraction of their ideal rate while eluding detection [22].

3.3 Detailed low-rate attacks classification

One of the biggest problems facing IT security professionals is reaching a consensus on a universal taxonomy on which to base the classification of cyberattacks. However, such taxonomies can be based on parameters that are completely independent of each other –such as the nature of the attack, the type of fundamental feature of computer security that it affects, or the way in which it is carried out and propagated. For this reason, a similar phenomenon occurs when it comes to classifying low-rate attacks in a more exhaustive way. Therefore, in this work we will proceed to explain the two classifications that are considered most relevant to understand the functioning of low-rate DoS attacks in a holistically way.

3.3.1 Low-rate DoS based on traffic flow parameters

Zhang et al. proposed in [23] a simple and precise classification that is briefly presented in this section. As previously mentioned, the goal of this work is to compare and determine the most effective detection method for low-rate DDoS attacks. In order to achieve it, we will filter network traffic by flows. A flow is defined in [24] as a set of IP packets passing an observation point in the network during a certain time interval, which share a set of

properties. These common properties may include packet header fields, such as source and destination IP addresses and port numbers, packet contents and meta-information.

Moreover, flow records are defined as information about a specific flow that was observed at an observation point, which may include both intrinsic properties of a flow –IP addresses and port numbers– and measured properties –packet and byte counters. They can be imagined as rows in a typical database, with one column per property [25].

For the purpose of this work, we will take into consideration that a flow is determined by a 5-tuple as stated by Zhang et al. [23]. Any combination of source address, source port, destination address, destination port and protocol can be used to group traffic packets in flows. We will only choose source IP address for the experimental tests of information theory-based detection methods specifically for sake of simplicity. We will define each flow with F_i.

A distributed low-rate attack consists of multiple attack flows $F_1, F_2, ..., F_n$ which come from different machines over the Internet. Four parameters are needed to describe a low-rate DoS attack flow:

1. T_a: attack period
2. T_b: attack burst width
3. R_b: attack burst rate
4. s: starting time of the attack flow

As we have defined the concept of flow, we can describe a distributed low-rate attack by using four parameters:

1. n: total number of flows in the attack
2. g: number of attack flow groups
3. m: number of members in an attack flow group
4. σ: constant starting gap between consecutive flow groups

Therefore, low-rate attacks can be classified in four categories as shown in Figure 3.2 depending on how T_a, T_b and R_b are being distributed amongst multiple flows:

1. **Attack Frequency Intensification (AFI)**: the aggregate attack period (T_a) is equally distributed amongst n flows. This way, the attack frequency of the aggregate flow is intensified by n times, in comparison to the frequency of each attack flow (Figure 3.2.a).

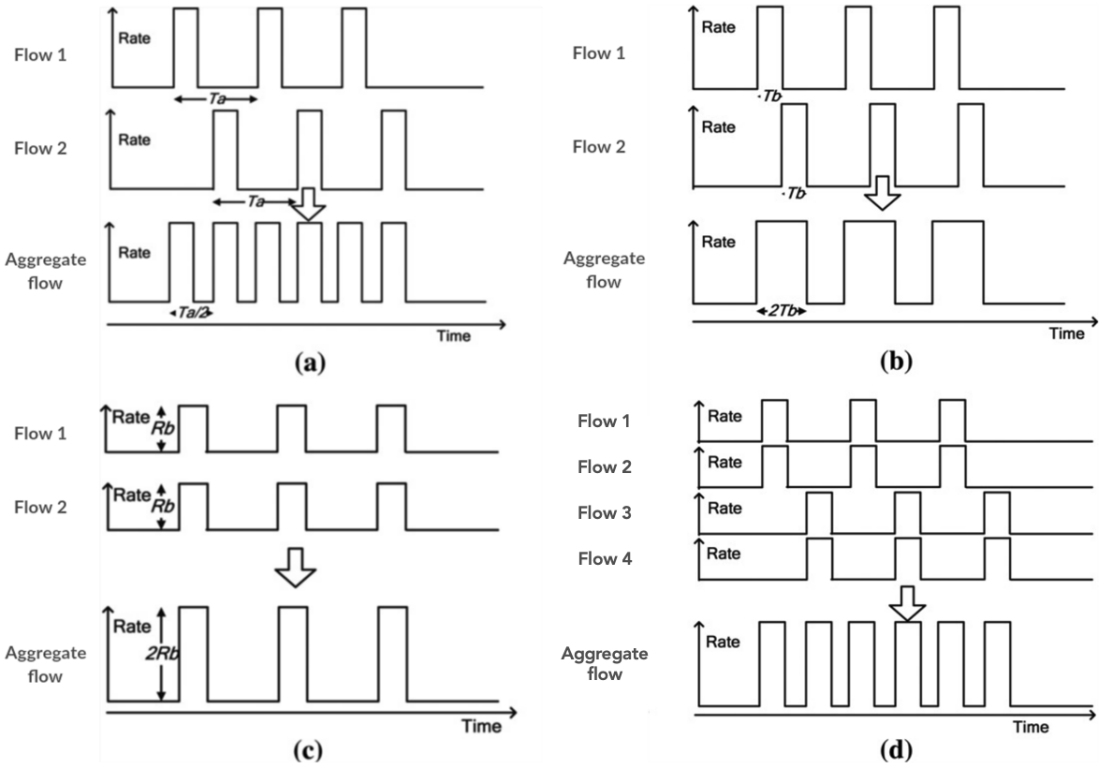


Figure 3.2: Low-rate DDoS attack classification by Zhang et al. (a) Attack Frequency Intensification (AFI), (b) Attack burst Width Intensification (AWI), (c) Attack burst Rate Intensification (ARI), and (d) Mixed Intensification (MI). The unit for Rate is Bytes/s and the unit for Time is second.

2. **Attack burst Width Intensification (AWI)**: the burst width of the aggregate flow is equally distributed amongst n flows. In other words, an attack burst of a flow is immediately followed by a burst from another flow. That is, the attack burst width of the aggregate attack flow is intensified by n times (Figure 3.2.b).

3. **Attack burst Rate Intensification (ARI)**: n flows start at the same time, and the burst rate of the aggregate attack flow is intensified by n times (Figure 3.2.c).

4. **Mixed Intensification (MI)**: any combination of the previous three (Figure 3.2.d).

We consider this classification to be certainly interesting since it differentiates between period, width and rate –common features in all kinds of signals– and help us to better understand the nature and behavior of low-rate attacks compared to high-rate ones.

3.3.2 Low-rate DoS based on techniques used to negatively impact the availability of the victim

During the exhaustive research process carried out for the development of this work, one of the main problems we encountered was the diversity of network implementation proposals for the deployment of a low-rate attack based on the mechanism for sending attack packets and therefore, the choice of a single type of implementation of attack submission rate approach that could meet the requirements of all detection methods to be compared.

Focusing on the exploitation of the intrinsic vulnerability of the control mechanisms that manage the TCP congestion and taking as a reference the different patterns of sending malicious packets adopted in various articles relating to the object of study, a clear division can be established between **pulsing attacks** and **constant attacks** for carrying out DoS attacks, both high-rate and low-rate.

Pulsing attacks are characterized by periodically sending burst attack packets over short periods of time repeatedly whereas **constant attacks** are performed by sending constant traffic but with a rate of packets per second that would not be identifiable at a glance if we divided the different existing flows in a network and superimposed them in a visual representation.

In **pulsing attacks**, burst periods are normally used by attackers to explore the homogeneity of the minimum retransmission timeout (RTO) while performing pulsing attacks. As described in Kuzmanovic's research, for an effective execution of these type of attacks, periodic on-off "square-wave" shrew attacks are defined, consisting of short, maliciously-chosen-duration bursts that repeat with a fixed, slow-time scale frequency [22].

The square-wave stream pattern can be observed in Figure 3.3. The effectiveness of this type of attack resides only in the attacker being able to retransmit the packets periodically following this on-off pattern, so that it collides with the legitimate packets at the moment of reaching the attacked victim. According to the implementation of the TCP protocol, if a legitimate user's packet cannot be delivered correctly, and is therefore considered a packet loss, the TCP flow will enter a timeout hold and attempt to send a new packet RTO seconds later. This mechanism is used as an attempt to avoid congestion collapse. In case of additional loss, the RTO doubles with each subsequent waiting time.

Therefore, it is obvious to consider pulsing attacks very difficult to be successfully executed, especially when the victim is configured within a network with a wide bandwidth and thousands of simultaneous connections per second –such as a video-on-demand server– since the real impact that this attack would have on such a network would be so low that it could even be irrelevant to detect and mitigate.

Moreover, existing signature-based metrics for detecting pulsing attacks are based on finding an underlying pattern in the analyzed traffic, namely the burst –or pulse– period,

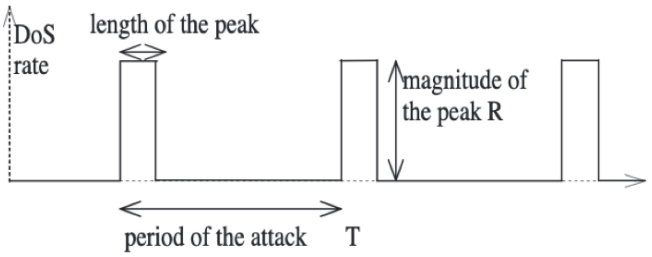


Figure 3.3: Underlying square-wave DoS stream on pulsing low-rate attacks as presented by Kuzmanovic in [22].

which is usually set to 1 second. However, the ineffectiveness of this method has recently been demonstrated in [26], as this metric does not take into account certain parameters of the network environment under study, such as traffic congestion while a low-rate attack is occurring. It is therefore necessary to develop anomaly-based detection methods for an effective detection of pulsing attacks.

On the other hand, the successful execution of **constant attacks**, based mainly on a constant sending of slow traffic at a rate that is very difficult to distinguish from the legitimate traffic rate, is achieved as long as the greater number of simultaneous connections towards the victim are kept open for as long as possible.

If we continue to consider the TCP protocol for carrying out low-rate attacks, there is a more particular denomination of constant attacks known as "low and slow attacks". These focus on the application layer and specifically, on the HTTP and TCP protocols for sending malicious packets. They usually target web servers, with the aim of tying up every thread with slow requests, thereby preventing legitimate users from accessing the service. This is accomplished by transmitting data very slowly, but just fast enough to prevent the server from timing out.

Within this category, the most widespread attack is known as Slowloris. Developed by Robert "RSnake" Hansen in 2009, his algorithm of attack allows a single machine to take down another machine's web server with minimal bandwidth by keeping many connections with the target web server open and holding them open for as long as possible. This way, affected servers will keep these connections open —receiving partial HTTP headers and waiting to receive the rest of the headers— filling their maximum concurrent connection pool, eventually denying additional connection attempts from clients as Slowloris sockets are tying up the thread.

Chapter 4

Comparative analysis of existing detection methods for low-rate attacks

Nowadays, there are plenty of proposals for DDoS attack detection methods, mainly high-rate attacks, but there are an increasing number of methods being developed for the specific detection of low-rate ones, due to the similar characteristics that low-rate traffic shares with legitimate traffic.

For the present work, two main methods have been selected, both based on metrics that can be implemented for experimental tests. In addition, as both employ a tolerance factor and the standard deviation of the sample as a function of the proposed metrics, performing an analysis of both has been considered to be quite representative for purposes of comparing results.

4.1 Detection based on information theory metrics

Low-rate attacks are far way more difficult to detect than disruptive ones as they submit attack packets periodically. To defy that hindrance, we introduce entropy as the basis of information theory metrics to accurately detect them.

The concept of information theory refers to the field of study of communication, quantification and storage of information. A key metric in information theory is entropy, which can be defined as the amount of uncertainty associated to a random variable or more accurately, as the amount of information gained by the observations of disordered systems. Other information theory metrics are mutual information, channel capacity, error exponents and relative entropy —most commonly known as Kullback-Leibler divergence. Since the calculation of information distance metrics requires that the two compared sets have the same number of elements —i.e., the same number of captured

network flows— for the present work we will mainly focus on entropy metrics for simplicity when generating traffic for experimental tests.

Back in 2011, Shannon's entropy and Kullback–Leibler's divergence metrics were assumed to be the most effective methods in detecting irregular traffic based on IP address or packet size distribution statistics [27]. However, the constant evolution of Internet traffic and the increasing amount of data transmitted over the network every second has made it necessary to investigate more effective methods to suit the varying characteristics of traffic. This is why two recently developed entropy metrics are presented along with Shannon's entropy, which the former ones have proven to be quite more effective on both real and malicious network traffic captured in 2007 than the latter one.

In information theory, entropies make up of the basis for distance measures among various probability densities. The development of the idea of entropy of random variables by Claude Shannon provided the fundamentals of information theory. Entropy and related information measures provide useful descriptions of the long-term behavior of random processes [28].

Comparing the rate of entropy of some sample of traffic flows to that of another sample of traffic flows provides a mechanism for detecting changes in the randomness. The entropy shows its minimum value 0 when all the items –source IP address for the present case study– are the same and its maximum value $log(n)$ when all the items are different. We will use the variation of entropy on compared traffic distributions based on source IP for DDoS detection. As we are mainly interested in measuring the entropy of flows over unique source IP addresses, then the maximum value that n can reach will be 2^{32} for IPv4 addresses.

The Shannon entropy equation provides a way to estimate the average minimum number of bits needed to encode a string of symbols, based on the frequency of the symbols. The entropy is higher as the information variable is more random. As a consequence, the greater certainty of the information variable is, the smaller the entropy is. Specifically, the generalized information entropy metric is very relevant in statistics as an index of diversity.

The **Shannon entropy** H is given by the formula 4.1, where p_i[1] is the probability of character number i appearing in the stream of characters of the message and b is the number of different signal levels that a character can adopt. As we are working with bits, the value of b will be 2 as bits can adopt either 0 or 1 values.

$$H(x) = -\sum_{i} p_i \cdot log_b(p_i) \tag{4.1}$$

[1] The p_i here denotes probability density for an i^{th} character. As probability density values lay in the range from 0 to 1, it means that logarithm functions will take on negative values. Since we talk about entropy, which cannot decrease in isolated systems (according to Second Law of Thermodynamics), it is mathematically more correct to put a "−" before the sum so that the value in the sum becomes positive.

More recent researches [15] have developed a novel metric that occurs to be more effective on the calculation of entropy of a system based on Shannon's entropy. One of them is the so-called **generalized entropy metric** given by Formula 4.2, which is a generalization of the Shannon's one, where p_i is the probability of the event x_i, with $x_i \in \{x_1, x_2, ..., x_n\}$ for $p_i \geq 0$. Adopted to our case, these events will be the different source IP addresses and p_i will be the density of traffic from said source address. This is a very interesting metric to use as we can obtain much better detection results by using generalized information entropy and adjusting the value of its integer order σ as we will demonstrate in Chapter 5.

$$H_\sigma(x) = \frac{1}{1-\sigma} log_2 \left(\sum_{i=1}^{n} p_i^\sigma \right) \tag{4.2}$$

Proof. For order $\sigma = 1$, the generalized entropy metric converges to the Shannon's one.

$$\lim_{\sigma \to 1} \frac{log_2 \left(\sum_{i=1}^{n} p_i^\sigma \right)}{1-\sigma}$$

As $\lim_{\sigma \to 1} log_2 \left(\sum_{i=1}^{n} p_i^\sigma \right) = \lim_{\sigma \to 1} log_2(1) = 0$, and $\lim_{\sigma \to 1} 1-\sigma = 0$, we obtain an indetermination as $\frac{0}{0}$. Therefore, we can apply L'Hôpital rule which states that the limit of the division of two functions will be same for the division of the derivates of the functions: $\lim_{x \to 1} \frac{f(x)}{g(x)} = \lim_{x \to 1} \frac{f(x)'}{g(x)'}$. Moreover, $log(f(x))' = \frac{f(x)'}{f(x)}$. The limit, after having derived both parts of the fraction, would remain:

$$\lim_{\sigma \to 1} \frac{\frac{\partial}{\partial \sigma} log_2 \left(\sum_{i=1}^{n} p_i^\sigma \right)}{-1} = \lim_{\sigma \to 1} - \frac{\frac{\partial}{\partial \sigma} \sum_{i=1}^{n} p_i^\sigma}{\sum_{i=1}^{n} p_i^\sigma}$$

We can derive p_i^σ as $\frac{\partial}{\partial \sigma} p_i^\sigma = log(p_i) \cdot p_i^\sigma$. The limit will be then:

$$\lim_{\sigma \to 1} - \frac{\sum_{i=1}^{n} p_i^\sigma \cdot log(p_i)}{\sum_{i=1}^{n} p_i^\sigma} = - \sum_{i=1}^{n} p_i \cdot log(p_i)$$

which corresponds to the Shannon's entropy formula. □

In the work published by Behal and Kumar in 2017 [21], reference is made to an improved entropy metric first enunciated in [29], known as **ϕ-entropy** and defined according to formula 4.3. This novel measure of information theory proves to be much more accurate in detecting low-rate attacks with precision.

$$H_\sigma{}'(x) = -\frac{1}{sinh(\sigma)}\left(\sum_{i=1}^{n}(p_i \cdot sinh(\sigma \cdot log_2(p_i))\right) \tag{4.3}$$

Proof. For order $\sigma = 0$, the ϕ-entropy metric converges to the Shannon's one.

$$\lim_{\sigma\to 0} -\frac{1}{sinh(\sigma)}\left(\sum_{i=1}^{n}(p_i \cdot sinh(\sigma \cdot log_2(p_i))\right)$$

As $sinh(\sigma) = 0$, multiplying every other element in the numerator, and $\sinh(\sigma) = 0$, we obtain an indetermination as $\frac{0}{0}$. Therefore, we can apply L'Hôpital rule which states that the limit of the division of two functions will be same for the division of the derivates of the functions: $\lim_{x\to 1}\frac{f(x)}{g(x)} = \lim_{x\to 1}\frac{f(x)'}{g(x)'}$. Since $(f(x)\cdot g(x))' = f(x)'\cdot g(x) + f(x)\cdot g(x)'$ and $sinh(f(x))' = cosh(f(x))\cdot f(x)'$, the limit, after having derived both parts of the fraction, would remain:

$$\lim_{\sigma\to 0} -\frac{\sum_{i=1}^{n} p_i \cdot cosh(\sigma \cdot log(p_i)) \cdot log(p_i)}{cosh(\sigma)} = -\sum_{i=1}^{n} p_i \cdot log(p_i)$$

which corresponds to the Shannon's entropy formula. □

Very different network traffic scenarios can provide the same entropy value, as this is absolute regardless of the characteristics of the context in which it is calculated. Therefore, we will need to know the normalized standard deviation of the network under study from a series of legitimate traffic samples. A considerable difference from the standard deviation of the traffic analyzed at a given time with respect to the value for the traffic modeled as legitimate will allows us to classify this certain traffic as malicious.

For the entropy-based detection algorithm, the s standard deviation of entropy in legitimate traffic is defined as shown in Formula 4.4:

$$s_N = \sqrt{\frac{\sum_{i=1}^{n}(H_N - H_i)^2}{n}} \tag{4.4}$$

being H_N the arithmetic mean of entropy calculated from n samples of legitimate traffic captured and H_i the entropy calculated from each one of said samples.

Based on the specific features of each network on which the detection method is deployed, a tolerance factor α must be tuned, which will allow the model to be stressed as much as possible so that detection would be as effective as possible, obtaining the lowest value rate of false positives.

With this tolerance factor, we define the distance gap $D_H(\alpha, \Delta t)$ in Formula 4.5 as the difference between the entropy of the analysed traffic and the average entropy of the legitimate traffic adjusted according to the tolerance factor $\alpha \in \mathbb{R}$ and the standard deviation s_N for legitimate traffic, for a sample in period t.

$$D_H(\alpha, \Delta t) = |H_C - H_N| - \alpha \cdot s_N \tag{4.5}$$

Therefore, the detection system set in Formula 4.6 will produce an affirmative detection of malicious traffic when the value of this distance gap is greater than or equal to zero, and would consider the traffic as legitimate for a value less than 0:

$$I_H = \begin{cases} 1, & D_H(\alpha, \Delta t) \geq 0 \\ 0, & D_H(\alpha, \Delta t) < 0 \end{cases} \tag{4.6}$$

4.2 Detection based on Expectation of Packet Size (EPS)

In October, 2017, a group of researchers from the Wuhan Polytechnic University, in China, proposed a new detection method based on a fundamental parameter of network traffic flows, the packet size [30]. Thus, instead of determining the existence of an underlying attack on the network from the entropy calculation –which has been proven to be quite limited as the difference between the entropy of legitimate traffic and malicious traffic is so small that it can lead to a high percentage of false positives– these researchers propose a method based on the expectation of packet size.

As explained earlier, DoS attack detection methods have traditionally been classified into two major categories: signature-based and anomaly-based methods (Figure 2.5). The former have quickly become obsolete as they analyze traffic distribution against a limited set of patterns, which does not lead to effective detection for low-rate attacks since attack traffic behavior at the network level is very similar to that of legitimate traffic.

If we analyze the characteristics of legitimate traffic, we observe that the size of packets varies depending on the protocol and the type of user information. However, legitimate traffic, which includes communication-protocol packets and user-data packets, follows a predictable packet size in normal situations.

Most communication-protocol packets have a predefined packet size in order to increase the effectiveness and facilitate detecting them by the hosts of a network. In addition, said packet size is usually small.

On the other hand, the size of user-data packets is usually larger, and can easily reach the Maximum Transmision Unit (MTU). The MTU is the maximum packet size that the data-link layer can transmit and is usually set to 1500 bytes in Ethernet v2 protocol.

However, when a low-rate DDoS attack is occurring, the packet size of a single legitimate packet, regardless of whether the communication-protocol packet or user-data packet is included, is very difficult to detect as stealthy [30], as each packet is a legitimate packet but the aggregation of them exhibits abnormal statistical deviations [31] and results in a low-rate DDoS attack.

As with the information theory-based detection method explained above, we are going to classify traffic into flows according to the same parameters –source IP, destination IP, source and destination ports, protocol used, number of packets sent and total bytes sent.

In paper [30], x_i is taken at the i^{th} flow for the selected analysis time range, c_i is taken as the number of packets sent for the flow x_i, and l_{ik} the size of packet k of flow i, which would imply the need to operate with an array where each row represents a flow and each column represents each of the packets received for each flow.

From these parameters, the mean of the packet size for the flow i, can be named $\overline{l_i}$ which follows the Formula 4.7:

$$\overline{l_i} = \frac{1}{c_i} \sum_{k=1}^{c_i} l_{ik} \tag{4.7}$$

Taking as a reference the number of packets in each flow, we can say that the probability of a flow x_i is defined in Formula 4.8 as:

$$p(x_i) = \frac{c_i}{\sum_{i=1}^{n} c_i} \tag{4.8}$$

Therefore, knowing that the sum of all probabilities will be equal to 1 and that the average packet size per flow is represented in bytes, the expectation of packet size (EPS) is defined in Formula 4.9 as:

$$EPS = \sum_{i=1}^{n} p(x_i) \cdot \overline{l_i} \tag{4.9}$$

However, the value of the mean of packet size $\bar{l_i}$ cannot be considered an absolutely representative value since the repetition of the same value n times has as average the same as if we take only the same number of values of the extremes. Therefore, we need to know the mean standard deviation s of the packets as a function of the EPS in order to determine whether there is a specific number of flows in the traffic that follow a given behaviour, which would lead into an excessively high value of s that would indicate that a low-rate attack is taking place. Specifically, the standard deviation measures how far away each of the values we have calculated the mean with are from said mean. In Formula 4.10 the standard deviation s for a certain traffic sample is defined as a function of the EPS value:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(EPS - \bar{l}_i)^2}{n}} \tag{4.10}$$

The fact that some artificially generated low-rate traffic has established patterns for certain traffic parameters that would be random in a real scenario, is what allows us to distinguish malicious flows from legitimate ones.

Specifically, we can take a series of samples of legitimate traffic and calculate their EPS and standard deviation, determining those values that should be considered normal. We can then use these results as a premise to directly detect anomalies in network traffic if we compare it with EPS and the standard deviation of traffic in real time.

Similarly as explained in the previous section, researchers [30] define $D(\alpha, \Delta t)$ as the distance gap between the metrics calculated for legitimate traffic and current traffic, considered under attack, for sample in period t. Specifically, they define $D_{EPS}(\alpha, \Delta t)$ as shown in Formula 4.11:

$$D_{EPS}(\alpha, \Delta t) = |EPS_N - EPS_C| - \alpha \cdot s_C \tag{4.11}$$

where $\alpha \in \mathbb{R}$ is a tolerance factor determined according to the features of the network to analyze.

However, due to the features of the programmatically simulated traffic not being adapted with sufficient certainty to the features of actual traffic, during the experimental phase (detailed in Chapter 5) it was observed that the behavior of the detection system does not operate as expected when $\alpha > 5$. Therefore, for simplicity of the present work, this parameter has been determined to be $\alpha = 1$.

Finally, the detection system is activated when it detects that $D(\alpha, \Delta t) \geq 0$, i.e. as indicated in Formula 4.12:

$$I_{EPS} = \begin{cases} 1, & D_{EPS}(\alpha, \Delta t) \geq 0 \\ 0, & D_{EPS}(\alpha, \Delta t) < 0 \end{cases} \tag{4.12}$$

4.3 Comparative analysis results

In order to verify the effectiveness of their proposed methods, the authors of both methods [21], [30] tested real data from datasets available for research purposes under request. Specifically, both use the attack profile of CAIDA datasets that contains approximately one hour of anonymized traffic traces from DDoS attacks that took place on August 4 and 22, 2007, for entropy method and EPS method respectively [32], [33].

On the other hand, for the profiling of legitimate traffic parameters, the EPS-based detection method makes use of real anonymized data gathered from a Fast Ethernet link

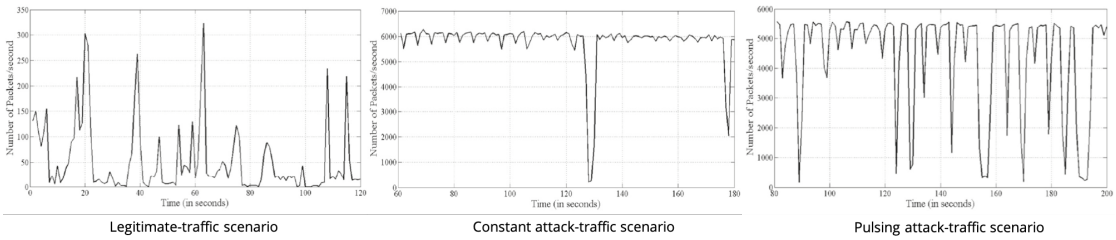


Figure 4.1: Both legitimate and attack traffic scenarios used for testing of the information theory-based detection method by Behal and Kumar in [21]. Constant and pulsing attacks were already explained in Section 3.3.2.

that connects the Widely Integrated Distributed Environment (WIDE) backbone network in Tokyo [34]. Information theory-based metrics (Shannon's entropy, generalized entropy and ϕ-entropy) use MIT Lincoln dataset [35], a custom designed DDoS testbed, FIFA 1998 World Cup dataset [36] and underlying legitimate traffic from CAIDA dataset [33]. All these datasets were used for testing the three aforementioned information theory-based metrics.

It can be observed that the pulsing traffic used by both authors for the experimentation follows the typical features of high-rate attacks, reaching a packet size rate much higher than the average rate of legitimate traffic. However, Chapter 5 implements a low-rate attack traffic scenario with a more normalized packet size using the ns-3 network simulator.

Among all the DDoS attack detection methods that have been discussed to date, the two presented methods in Sections 4.1 and 4.2 have been expressly chosen as both use an α tolerance factor to calibrate the detection algorithms. The research carried out on this factor has led us to understand the importance of performing a very precise tuning process to reduce the rate of false positives as much as possible without affecting the functionality of the detection algorithm.

In Figure 4.1, we can observe the traffic scenario used for testing the novel information theory-based metrics developed by Behal and Kumar in [21]. Since the traffic profiles of the two studies are very similar, we can accurately compare the effectiveness of each one of them and determine experimentally which detection results to be more accurate. Additionally, in Figure 4.2, we can observe the traffic scenario used for testing the effectiveness of the EPS detection method presented by Zhou et al. [30]. All presented datasets contain traffic data extracted from 2007, so it is clear that the behaviour of the Internet traffic has evolved since then and the effectiveness of the studied detection methods should be measured on more up-to-date traffic captures.

As mentioned above, constant attacks generate ilegitimate traffic at a constant low-rate, whereas pulsing attacks periodically send bursts of packets to victims. Either

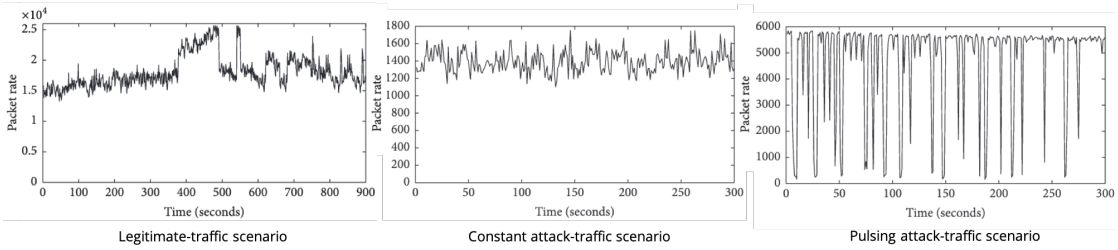


Figure 4.2: Both legitimate and attack traffic scenarios used for testing of the EPS detection method by Zhou et al. in [30]. Constant and pulsing attacks were already explained in Section 3.3.2.

approach usually employs a fixed and small packet size, more closely resembling the size of communication-protocol packets than user-data packets. Therefore, the probability of receiving small packets of a given size is much greater for attack traffic than for legitimate traffic. The latter are distributed following a Poisson-Pareto process.

When developing a comparative matrix that analyzes the effectiveness of different detection methods, four essential quadrants must be considered:

- True Positive (TP): rate of attack flows that are effectively reported as being attacks.
- True Negative (TN): rate of not attack flows that are effectively considered legitimate.
- False Positive (FP): rate of not attack flows that are mistakenly reported as being attacks.
- False Negative (FN): rate of attack flows that are mistakenly considered legitimate.

In this way, any detection method aims at increasing the value of TP and reducing to the maximum the value of FP, although it also may affect the performance of the network since the 100% of the malicious flows are not detected, as shown in Figure 4.3. In the case of ϕ-entropy, it is observed that a null positive phasing rate is obtained from the tolerance factor $\alpha = 2$, although the effective detection rate only reaches 75% of the cases, which means that 25% of the malicious flows are not correctly detected as such. If we take into account that of the information theory metrics presented, this was the one that has been shown to be the most effective, it is trivial to consider that Shannon's entropy and generalized entropy will achieve even lower detection rates. This is of critical importance because classifying legitimate flows as attackers may have a much greater impact for the organization, since this implies that the connection of legitimate users to the service is blocked, causing a deterioration in the Quality of Service (QoS) provided.

Two of the most important factors that have to be taken into consideration in network design are service and cost. Service is essential for maintaining customer satisfaction. Cost is always a factor in maintaining profitability [37]. Therefore, making an effort towards granting a quality service is crucial.

In this sense, the term Quality of Service, in the field of networking, refers to control procedures that can provide a guaranteed level of performance to data flows in accordance to requests from an application/user using the network [37]. Our study is deeply related with QoS as we focus on the idea that, in business terms, degrading attacks are far way more dangerous to certain companies than disruptive ones. For example, banks or social network platforms would suffer more impact if facing a disruptive DDoS attack as transactions or immediate communications could not be carried out, resulting in the loss of millions of euros for the former and on the switch to another platform for daily use for the latter.

However, companies that offer entertainment services such as Netflix or Spotify will be considerably more beaten by a degrading DDoS attack, since a service whose performance has been considerably degraded from the expected may result in a customer dropping out said service.

The main reason for employing the standard deviation in function of the EPS as part of the detection equation (Formula 4.11) relies on the fact that, although it has been determined that the EPS of legitimate traffic is usually greater than the EPS of low-rate attack traffic, said EPS in a network may dynamically vary and the false negative rate may increase due to the stochastic nature of network traffic [30]. The standard deviation is used for measuring the degree of deviation of the attack traffic packet size from its expectation. Thus, it is trivial that the standard deviation of attack traffic flows will be small, as these packets are generated following a pattern to hide among legitimate traffic flows but establishing a fixed packet size that standard deviation can help us detect.

On the other hand, the entropy-based detection system makes use of standard deviation s based on legitimate traffic. There is a reason why the two methods differ in the detection equations $D(\alpha, \Delta t)$. While the entropy-based method calculates the standard deviation s of the traffic considered legitimate, the EPS-based method calculates the standard deviation s of the traffic under study –the one presumably under attack. This is because the value for $H_N < H_C$ whereas $EPS_C < EPS_N$. This way, we need to contrast the gap between the values for attack and legitimate traffic with the smaller standard deviation in order to successfully detect a change in the behaviour of the network under study.

Therefore, it is certain that, for anomaly detection and regardless of the detection method used, the distance gap between the attack packets and the legitimate ones should be as high as possible to increase the detection sensitivity. However, the fact that this space is too large can lead to a high number of false positive detection since an appropriate tolerance factor has not been selected for the network under study. On account

Tolerance factor	ϕ-Entropy		Expectation of Packet Size (EPS)			
	Detection rate (%)	False positive rate (%)	Detection rate (%)		False positive rate (%)	
			Constant	Pulsing	Constant	Pulsing
$\alpha = 1$	85	16	96	99,98	4	0,04
$\alpha = 2$	77	1	98	99,98	2	0,04
$\alpha = 3$	75	0	98	99,99	1,4	0,04
$\alpha = 4$	75	0	98	99,99	1,35	0,04
$\alpha = 5$	75	0	99,5	99,99	1,3	0,04

Figure 4.3: Performance comparative table showing the experimental results carried out by the authors of the two detection methods under study.

of the nature of Internet being schotastic, this tuning process for the tolerance factor must be firstly performed on all networks in which the detection method would be implemented, since the network features can vary greatly from one to another and therefore, the profiling of what is considered legitimate traffic.

For both detection methods, Figure 4.3 shows, for values of the incremental tolerance factor, the percentage of the successful detection rate, i.e., the total number of malicious flows detected as such, and the false positive rate, i.e., those legitimate flows that the algorithm classified as malicious.

Therefore, it can be concluded that the low-rate attack detection method based on expectation of packet size is far more effective, both in detecting almost all malicious attack flows and in reducing the number of legitimate flows considered as attackers. On Chapter 5, we will test the performance of both anomaly-based methods on network-based traffic and host-based traffic in order to determine the effectiveness of each one depending on the source (Figure 2.5) under study.

Chapter 5

Experimental results

We now proceed to study the efficiency of the metrics explained on the previous chapter –the three aforementioned entropy metrics and the EPS one. For this purpose, an experimental study has been carried out on a network scenario where low-rate traffic was simulated. For the set experimental scenario, we will apply both detection methods and assess their effectiveness. On the assumption that network traffic has the implicit feature of self-similarity, the results we obtain on a small network should be as correct as those calculated on datasets of much larger networks.

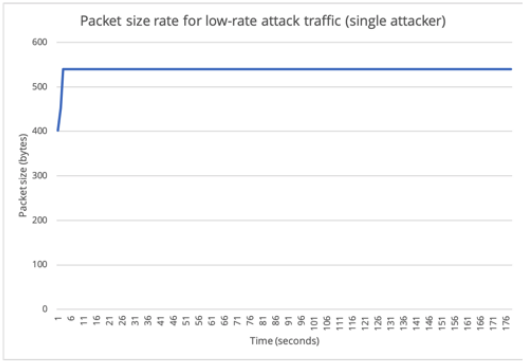
5.1 Experimental setup

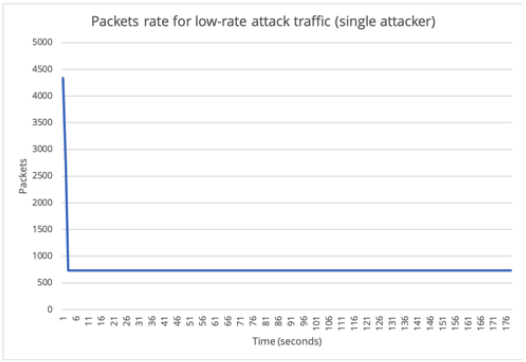
5.1.1 Network architecture

For the implementation and demonstration of the comparative analysis carried out on the different methods of detection of low-rate attacks, the initial idea was to simulate a whole network with legitimate nodes, attacking nodes and the affected server, whose availability would be affected by malicious packets from the attacking nodes.

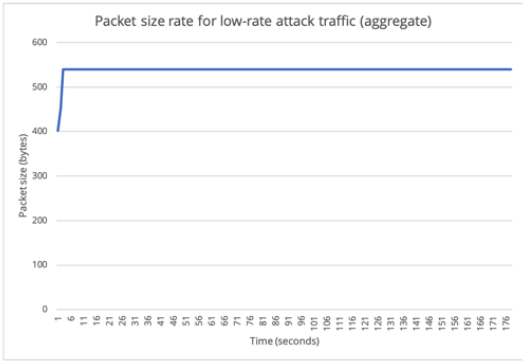
However, when testing the detection methods implemented on the traffic generated with the simulator, the results were quite different from the expected behavior. This was mainly due to the stable behavior of the simulated traffic, which follows an established pattern and it is practically impossible to simulate programmatically a random behavior similar to the behavior of real traffic. To overcome this limitation, only attack traffic was simulated, which as previously explained, could be expected to have a behavior determined by the properties of low-rate traffic, and then it was merged with actual traffic captured from a particular network.

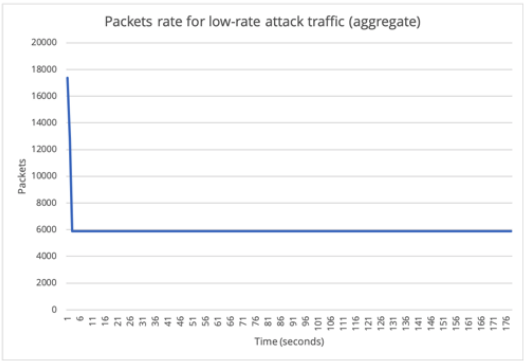
Out of the two existing methods for carrying out DDoS attacks that are object of study (and explained in previous chapters), the method of pulsing attacks was firstly chosen due to the ease of implementation. In this approach, the attacking nodes followed the pattern of sending, from set time to time, packets that interfered with the legitimate traffic, so that the legitimate nodes could not effectively connect to the server.





Packet size and packets rates for single attacker





Packet size and packets rates for the aggregation of four attackers

Figure 5.1: Packet size and number of packets rates for single attacker traffic and four attackers aggregate traffic generated.

Nevertheless, by working on a nanosecond scale to make the impact of low-rate attacks tangible on a small network without these becoming high-rate attacks, it was observed that simulated malicious traffic rather followed the pattern of a constant attack. Figure 5.1 shows the packet size rate and total number of packets as a function of time for ns-3[1] simulated attack traffic. On the one hand, it can be observed that the packet size remains constant, regardless of the number of attackers generating malicious traffic at a time. On the other hand, the number of packets sent to the network increases with the addition of new attackers, following a pattern of constant attack.

The implementation proposed by Samvid et al. interconnected the different nodes of the network by following a point-to-point topology, as can be seen in Figure 5.2. This topology, at the data layer level, makes it possible to physically and directly connect two

[1] ns-3 is a network simulator developed on C++ that has been used to generate the attack traffic in a controlled environment. It will be later explained on this chapter.

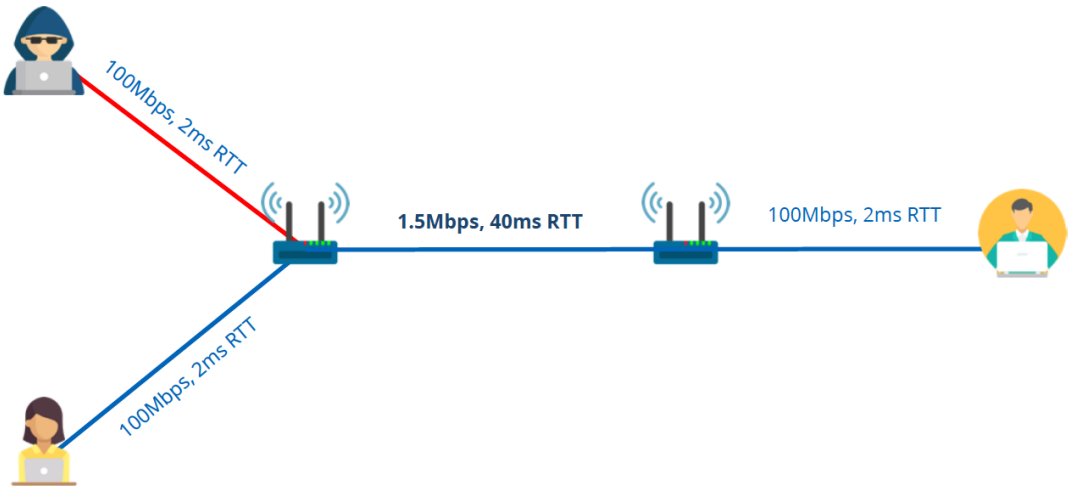


Figure 5.2: Low-rate attacks network topology implementation with ns-3 simulator proposed by Samvid Dharanikota et al. [38]

nodes of a network to each other. In this approach, the low-rate attacks attempt to deny bandwidth to TCP flows while sending packets at sufficiently low average rate to elude detection by counter-DoS mechanisms. This architecture consists of a single bottleneck queue driven by n long-lived TCP flows with heterogeneous RTTs and a single DoS flow. The DoS flow is a periodic square-wave DoS stream. This way, periodic DoS streams are not utilizing TCP's exponential backoff mechanism but rather exploiting repeated timeouts.

However, for the experimental tests, we found the problem that we could not mix traffic with different topologies at the data link level, since legitimate traffic captured on a home network implemented the Ethernet topology and attack traffic simulated by ns-3 implemented the point-to-point topology.

In order to work with normalized traffic captures, since the acquisition sources of the different types of traffic are carried out at different times and networks, we proceeded to modify the timestamp of the simulated attack packets so that they coincide with the date and time when traffic was captured in a real network. We also modified the destination IP address of the victim so that it is the same for both legitimate and attack traffic.

With the legitimate traffic that we captured in a real network, we proceeded to filter it in order to keep only the TCP traffic whose origin or destination was the IP address of the attacked server. Legitimate traffic has been captured from a real home network. In addition, in order to test the effectiveness of the methods studied according to the source of the traffic (Figure 2.5), traffic has been captured at the victim host interface

# Sample	Shannon	GE (σ=2)	GE (σ=5)	GE (σ=10)	ϕ- Entropy (α=0.1)	ϕ-Entropy (α=0.25)	ϕ-Entropy (α=0.5)	ϕ-Entropy (α=0.95)	EPS
1	4,14584926	3,65199086	11,41443132	22,92735447	4,28835446	5,09254898	8,85330330	39,90428163	1369,18887404
2	3,99643934	3,13606155	8,68379632	17,36965582	4,14077288	4,95583516	8,76833065	39,69666973	1154,95669001
3	3,99058638	3,23236271	9,35332694	18,74446354	4,13285450	4,93766159	8,72874649	40,29788712	1211,92946945
4	3,78063906	3,47643804	11,80278331	24,12273169	3,88283178	4,45028488	6,95175538	24,23400841	1325,39821573
5	3,86151563	3,18719359	9,20129683	18,41301829	3,98705183	4,69265433	7,94241042	33,44161210	1406,04223890
6	3,76385619	3,50635267	12,14580879	25,43644257	3,86102720	4,39788691	6,71946388	21,74333550	1206,60957722
7	3,47907041	2,88041838	9,01414202	18,43978089	3,57872157	4,13673183	6,67149816	25,71454581	1197,59893316
8	3,27982278	2,91427013	9,71639712	19,95037850	3,35430372	3,76617264	5,55491937	17,38622092	1334,49964381
Average	**3,787222382**	**3,248135992**	**10,16649783**	**20,67547822**	**3,903239742**	**4,55372204**	**7,523803455**	**30,30232015**	**1275,777955**

Figure 5.3: Different entropies and EPS values calculated for 8 different traffic samples considered legitimate and captured in the victim host.

# Sample	Shannon	GE (σ=2)	GE (σ=5)	GE (σ=10)	ϕ- Entropy (α=0.1)	ϕ-Entropy (α=0.25)	ϕ-Entropy (α=0.5)	ϕ-Entropy (α=0.95)	EPS
1	3,37036424	2,00623016	5,24851451	10,49753031	3,53424240	4,49805418	9,72084711	75,43946696	758,88377475
2	3,38689517	2,07505815	5,53834913	11,08059484	3,54871608	4,49813487	9,58895919	70,81194495	703,28664600
3	3,58753286	2,35299211	6,33624031	12,67480261	3,75422764	4,73494319	10,06655921	78,63976667	747,12201651
4	3,71464104	2,85525356	8,74530623	17,65256383	3,86729844	4,76104623	9,54809498	69,60158745	946,57253363
5	3,38448816	2,39223620	6,79765603	13,61185083	3,52626581	4,36193501	8,94803013	70,60361965	944,74038913
6	3,57515061	2,69780754	8,24704372	16,64888820	3,71972199	4,56463000	9,05293976	63,23432088	680,69747448
7	3,19826943	2,18550654	5,96176045	11,92644990	3,31817781	4,01136827	7,55530900	46,14348153	696,83424038
8	3,32336955	2,34406193	6,39942206	12,80107889	3,44239755	4,12504747	7,51690338	41,59962487	900,01805539
Average	**3,44258888**	**2,36364327**	**6,65928656**	**13,36171993**	**3,58888096**	**4,44439490**	**8,99970535**	**64,50922662**	**797,26939128**

Figure 5.4: Different entropies and EPS values calculated for 8 different traffic samples considered legitimate and captured in a home network.

and at the network interface of the home router. The different entropies and EPS values calculated for 8 different legitimate traffic samples is shown in Figure 5.3 for host-based capture and Figure 5.4 for network-based samples.

From this legitimate traffic, several PCAP[2] files have been generated:

- PCAP for attack-only traffic: it stores all the malicious flows that send constant data towards the victim host.
- PCAP for legitimate-only attack, captured on host interface: it stores all the TCP and UDP traffic sent and received only by the victim server.
- PCAP for legitimate-only attack, captured on network interface: it stores all the TCP and UDP traffic sent and received in the home network where the victim host is connected. Not all flows captured interact with the affected machine.

[2]PCAP stands for *packet capture* and it is an application programming interface for capturing network traffic. Once the traffic has been captured with a PCAP library (as it could be libpcap, integrated in Wireshark, whose functionalities we will explain in Section 5.1.2), a file with .pcap extension can be generated and later analyzed with any network analysis tool.

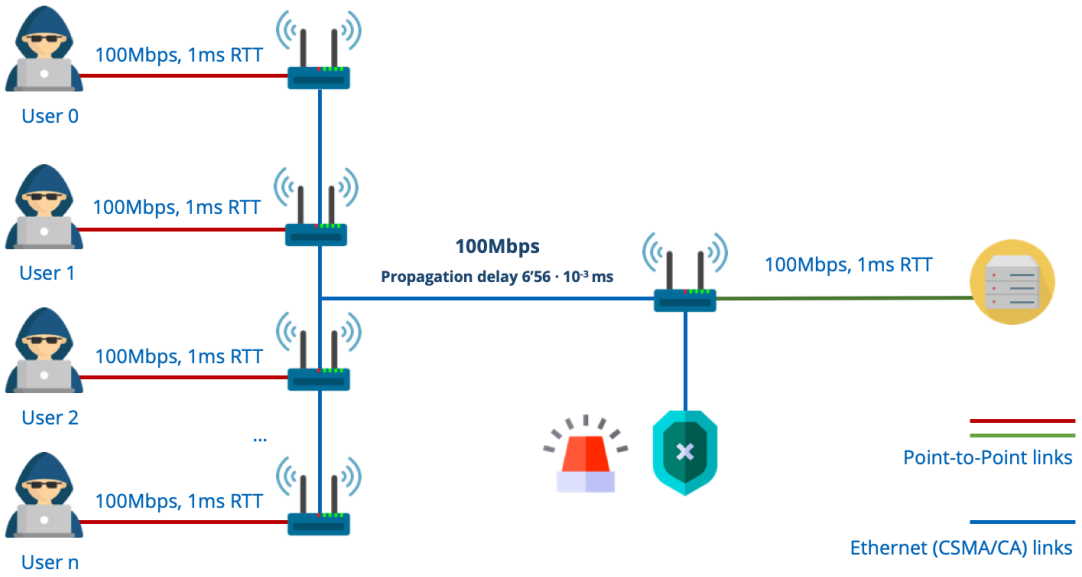


Figure 5.5: Low-rate attacks network topology implemented with ns-3 simulator.

- PCAP for legitimate host-captured traffic merged with attack traffic: it stores all malicious and legitimate flows towards the victim host. The are no other flows whose source or destination address is not the targeted machine.
- PCAP for legitimate network-captured traffic merged with attack traffic: it stores all malicious and legitimate traffic sent and received in the home network where the victim host is connected. Not all flows captured interact with the affected machine.

Specifically, the network topology implemented in ns-3 for the generation of low-rate attack traffic would remain as shown in Figure 5.5. For testing the performance of both methods depending on te percentage of attacking users generating malicious activity on the network.

5.1.2 Technologies and tools employed

ns-3 Network Simulator

The previously described network has been implemented programmatically using the ns-3 network simulation tool.

ns-3 [39] is a discrete-event open simulation network environment for Internet systems based on C++ programming language. ns-3 provides models of how packet data networks work and perform, and provides a simulation engine for users to conduct simulation experiments.

One of the main advantages of ns-3 is that users can emit and receive ns-3-generated packets on real network devices. It can also serve as an interconnection framework to add link effects between virtual machines. The main interest for using ns-3 simulator on our work is to study the behavior of a system under a DDoS low-rate attack in a highly controlled, reproducible environment, which allows us to implement the most convenient network architecture for testing the different implemented detection methods.

In addition, for the installation of project dependencies during the installation and implementation of the ns-3 simulator, Bake has been used. It is specifically an integration tool which is used by software developers to automate the reproducible build of a number of projects which depend on each other and which might be developed, and hosted by unrelated parties. It works under Linux operative systems and is implemented in Python.

Wireshark

Wireshark [40] is an open source network traffic and packet analyzer, or "sniffer", originally developed for Unix and Unix-like operating systems and released under the terms of the GNU General Public License. It uses Qt, a graphical user interface library, and `libpcap`, a packet capture and filtering library.

As a network packet analyzer[3], Wireshark captures network packets and displays packet data as detailed as possible. As a data capturing program, this tool understands and shows the encapsulation of different networking protocols. It can parse and display the fields, along with their meanings as specified by different networking protocols.

The main purposes of Wireshark tool are troubleshooting network problems, examining security issues, verifying network applications, debugging protocol implementations and learning network protocol internals.

Wireshark shares many characteristics with `tcpdump`. The difference is that it supports a graphical user interface (GUI) and has information filtering features. In addition, Wireshark permits the user to see all the traffic being passed over the network by putting network interface controllers into promiscuous mode (if supported by the network interface controller), so they can see all the traffic visible on that interface including unicast traffic not sent to that network interface controller's MAC address.

SiLK

The System for Internet-Level Knowledge, better known as SiLK, is a collection of traffic analysis tools developed by the CERT Network Situational Awareness Team (CERT NetSA) to facilitate security analysis of large networks. The SiLK tool suite supports

[3]A network packet analyzer is a measuring device used to examine what is going on inside a network cable.

the efficient collection, storage, and analysis of network flow data, enabling network security analysts to rapidly query large historical traffic data sets. The vast majority of the current code-base is implemented in C, Perl, or Python to be run in Linux and Unix-based systems.

The SiLK Analysis Suite [41] is a collection of command-line tools for processing SiLK Flow records created by the SiLK Packing System or already captured packet files such PCAP. These analysis tools read binary files containing SiLK Flow records and partition, sort, count and summarize these records.

Listed below are the commands used in the implementation of ETL[4] code for processing and cleaning the data captured with Wireshark:

- `rwcount` summarizes SiLK flow records across time. It counts the records in the input stream, and groups their byte and packet totals into time bins. This command splits each flow record into bins whose size is determined by an optional argument.
- `rwp2yaf2silk` is a script to convert a .pcap to a single file of SiLK Flow records, which is a necessary format to be able to apply a wide range of other calculations using SiLK commands.
- `rwcut` reads binary SiLK Flow records and outputs the user-selected record fields in a textual, bar-delimited (|) format.

Python

Python [42] is an open-source, interpreted, high-level and interactive programming language first released in 1991. It supports multiple operating systems and programming paradigms, including object-oriented, imperative, functional and procedural, and operates under the Python Software Foundation license.

The SciPy library has been used for the implementation in Python. It can be described as a Python-based ecosystem of open-source software for mathematics, science, and engineering. This library has a very complete set of packages for the treatment and analysis of data. Those packages that have been used in the development of this work are listed below:

- `NumPy`: fundamental package for scientific computing with Python. It enables the use of N-dimensional array objects and linear algebra in an efficient and high level form.
- `Pandas`: package that provides high-performance, easy-to-use data structures and, specially of application for our work, 1-dimensional data series.

[4]Extract, Transform, Load (ETL) is the general procedure of copying data from one or more sources into a destination system which represents the data in a different way than the source does.

- `Statsmodels`: package that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.
- `Matplotlib`: plotting package that provides publication-quality 2D plotting and more simple 3D plotting.
- `Scikit-learn`: complete package for the application of machine learning in Python, specially in the areas of data mining and data analysis.

5.1.3 Structure of code implemented

For the simulation of low-rate attack traffic, the code initially developed by Samvid Dharanikota, Sagar Bharadwaj and Adarsh Honawad, who carried out the implementation proposed by [22] in the ns-3 network simulator, has been adapted. In order to be able to subsequently mix the actual legitimate traffic captured with this generated traffic, the code has been adapted to include CSMA nodes that would simulate routers connected via Ethernet.

Simulated traffic with ns-3 with point-to-point protocol does not share the same link layer protocol as normal Ethernet and Wifi traffic on any home network. For this reason, when generating malicious traffic, it has been necessary to modify the network architecture initially proposed by Samvid et al. to include CSMA nodes that allow capturing packets in the same link layer protocol to later combine it with legitimate traffic and generate a PCAP on which to test the effectiveness of the methods studied.

In addition, the attack is initially performed by five attackers simultaneously following a low-rate Mixed Intensification (MI) pattern explained in Section 3.3.1. In an AFI approach, that traffic from some attackers is sent to the victim just few instants after the traffic sent by the first other attackers. As some of them starts the sending at the same time, the attack also follows an ARI approach, thus becoming a MI attack.

For the calculation of the mathematical formulas applied in the detection algorithms of both methods, several Python libraries have been used. This programming language has been chosen due to its high processing capacity.

Finally, for the conversion of captured traffic from PCAP format to CSV format so that it could be easily interpreted in Python, several SiLK suite commands programmed directly through a bash script have been used.

5.2 Results for information theory-based metrics

5.2.1 Design of the algorithm

For the implementation of the detection method based on entropy metrics, the same algorithm shown in Figure 5.6 has been followed for the testing of the different presented

entropies. Due to the impossibility of generating attack traffic in real time since it has been necessary to modify a series of characteristics of the packets after their generation with the ns-3 simulator –such as the timestamp or the source ports from which the attacking machines sent the malicious packets–, the experimentation tests have been carried out on a deferred basis.

5.2.2 Results obtained from experimentation

Figure 5.7 shows the different entropy values obtained for the analyzed traffic samples, both attack-only traffic, legitimate-only –for both host-based and network-based detection approaches– and the combination of both. It can be seen that there is a considerable difference between the entropy value of attack-only traffic versus legitimate-only traffic and, consequently, the combination of the two leads to an increase in the value of the total entropy of the network under attack.

For the comparison of the different network scenarios, we have chosen the ϕ-entropy for order $\sigma = 0.95$ as is the entropy metric that generates the largest distance gap (Figure 5.7).

A sample of legitimate traffic captured at the same moment than the other 8 samples used to calculated the standard traffic parameters has been merged with attack traffic generated with ns-3 simulator. In order to determine the necessary minimum percentage of attackers in the network for the detection method to be effective, six different samples of traffic have been generated for 4, 5, 6, 7, 10 and 20 attackers.

For the **network-based** detection approach, the ϕ-entropy-based detection method has demonstrate to be effective when there are at least **22% of malicious users** in the network performing an attack. On the other hand, for the **host-based** detection approach, the minimum threshold for detection is reached when **at least 65% of the users are malicious**.

5.3 Results for Expectation of Packet Size (EPS)

5.3.1 Design of the algorithm

For the implementation of the detection method based on EPS, the algorithm shown in Figure 5.8 has been followed. As happened with the information theory-based detection method, the experimentation tests have been carried out on a deferred basis.

5.3.2 Results obtained from experimentation

Figure 5.9 shows the values obtained for the EPS and the standard deviation of the different traffic samples analysed. In contrast to the entropy-based detection method, where it was necessary to achieve a minimum percentage of malicious users on the network, the EPS-based method does not.

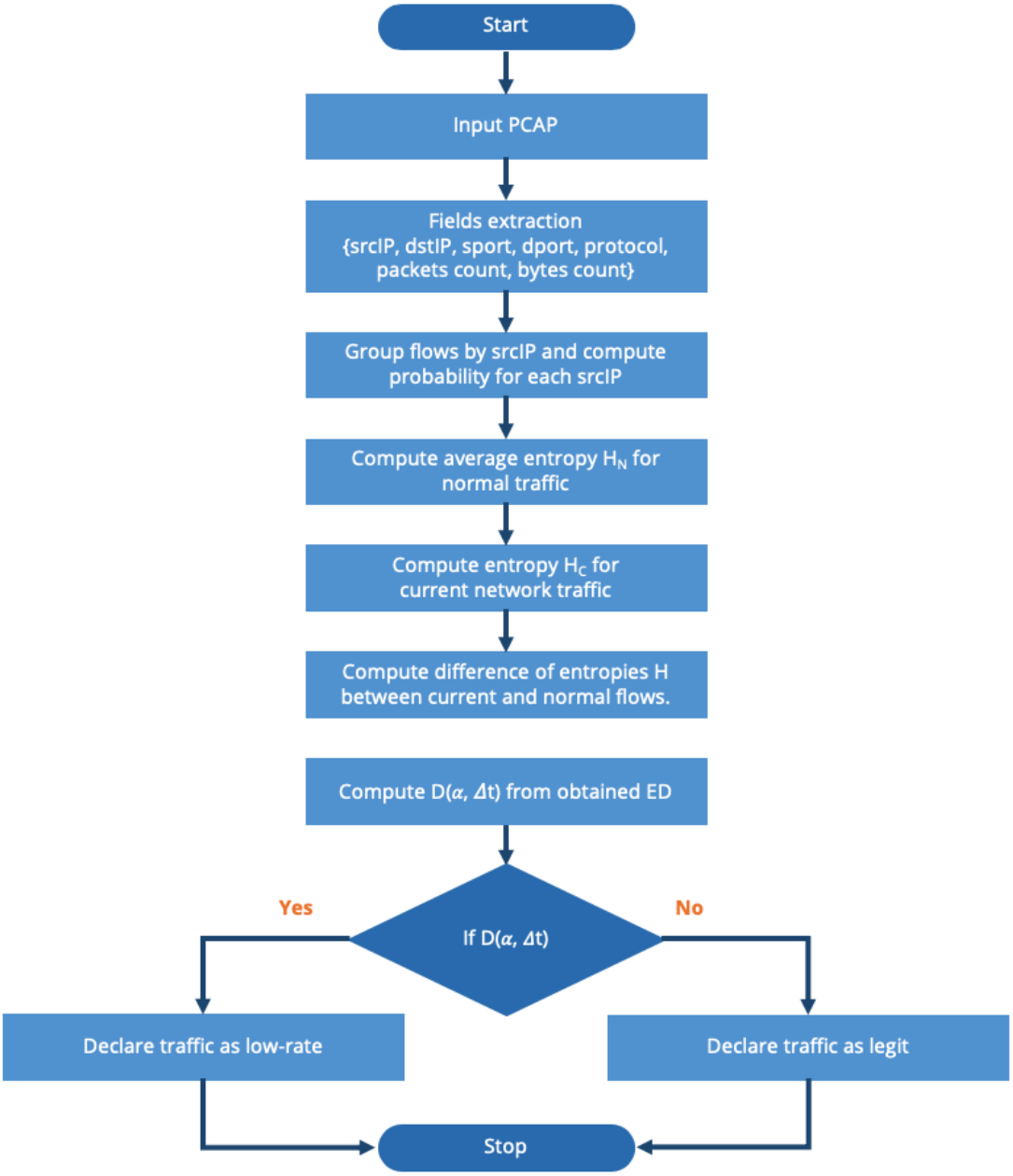


Figure 5.6: Proposed detection methodology by Behal and Kumar [21] adapted to our custom Python implementation.

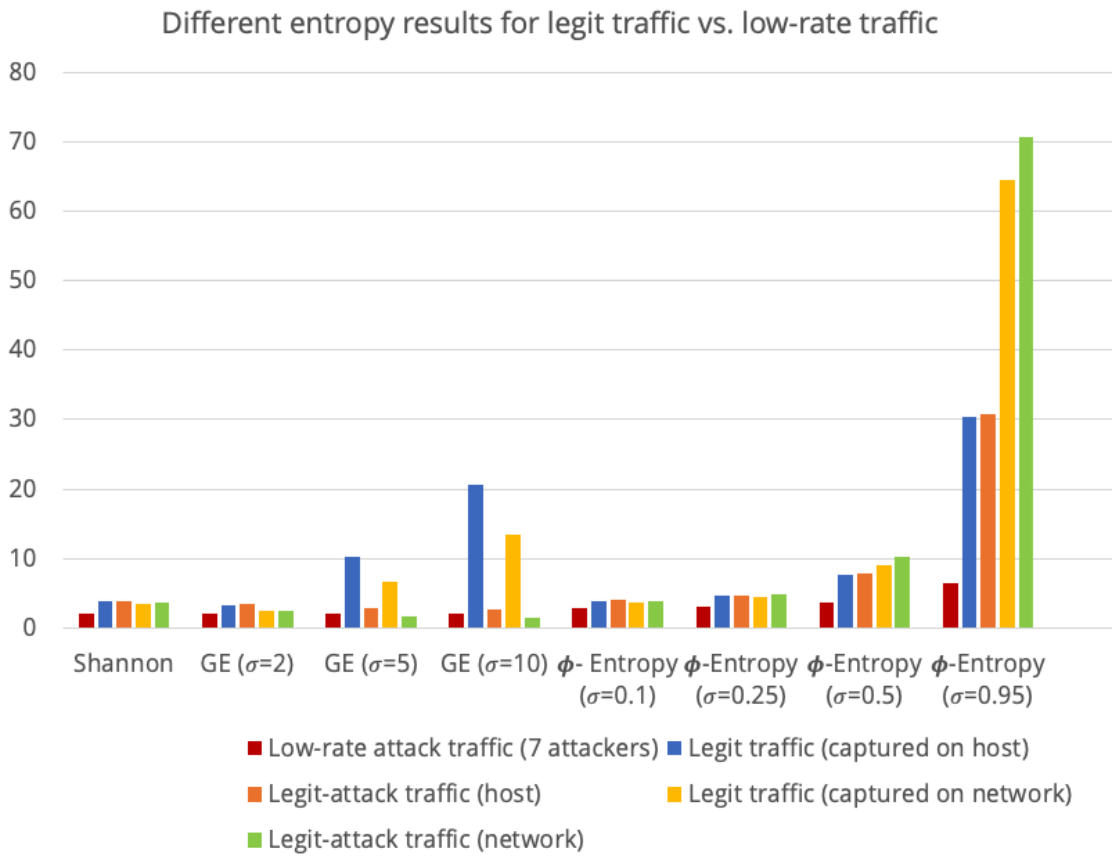


Figure 5.7: Comparison of entropy values obtained for attack traffic, legitimate traffic and merge of both for different values of σ.

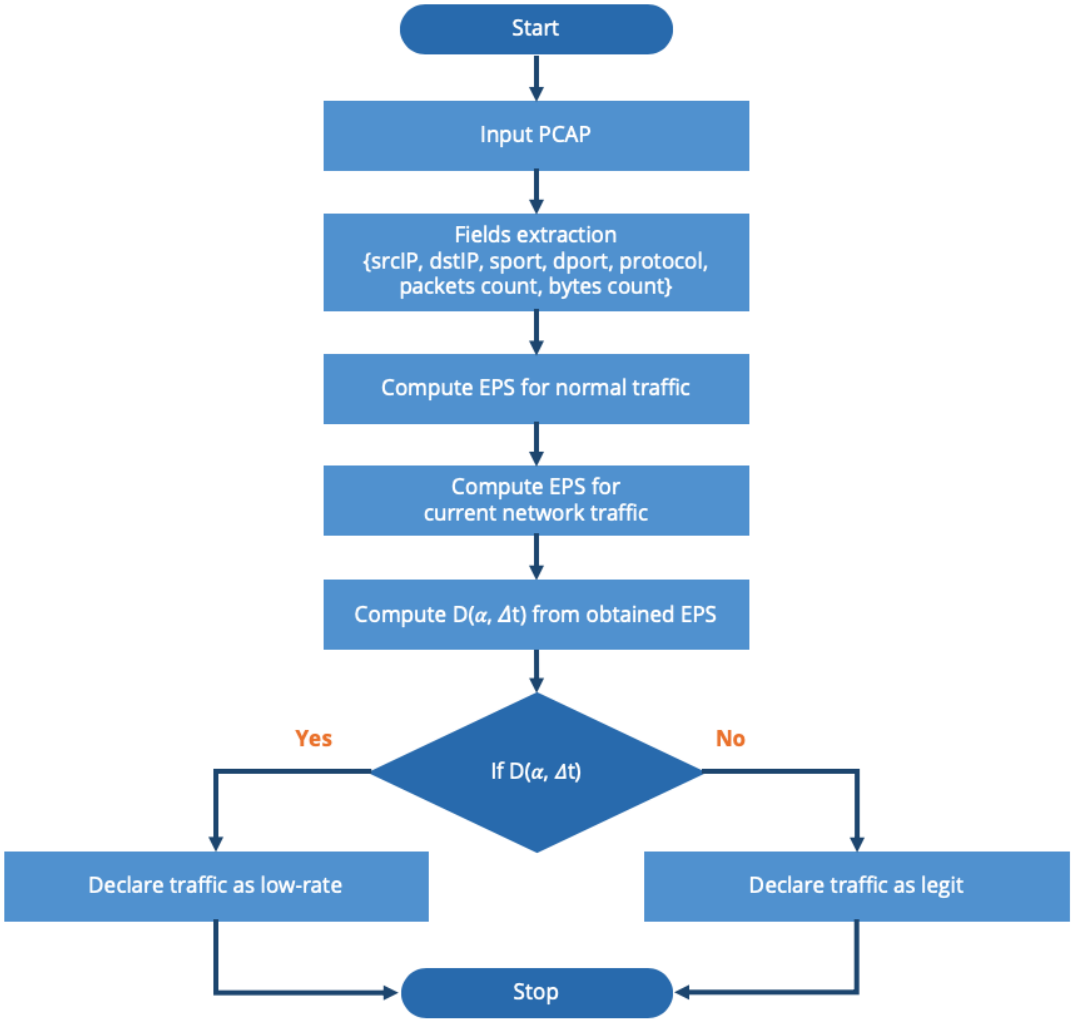


Figure 5.8: Proposed detection methodology by Zhou et al. [30] adapted to our custom Python implementation.

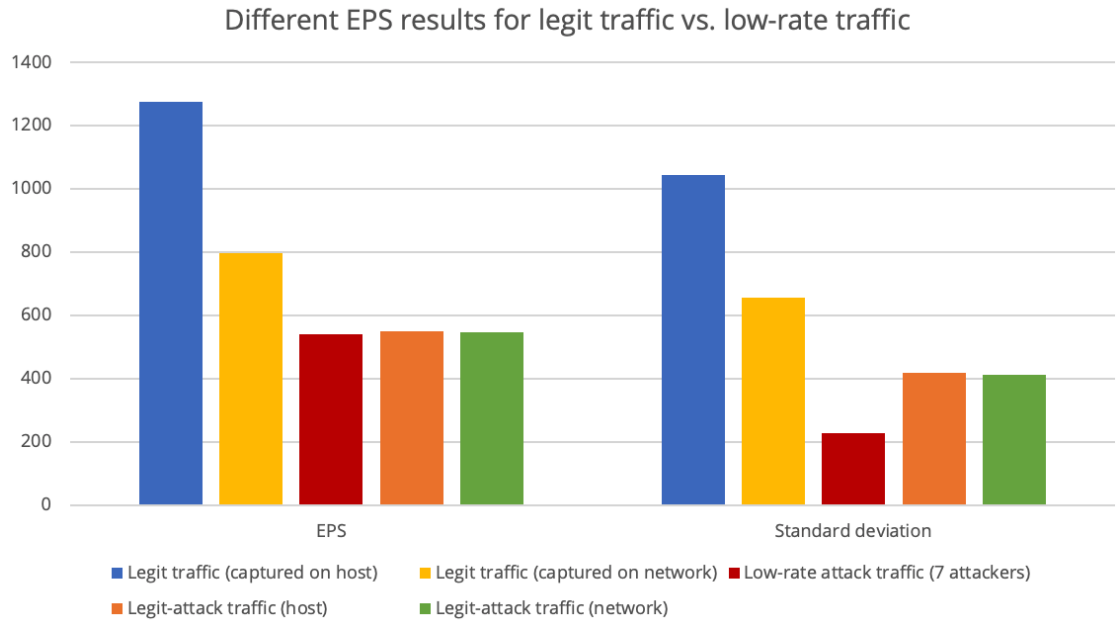


Figure 5.9: Comparison of EPS standard deviation values obtained for attack traffic, legitimate traffic and combination of both.

However, a considerable difference in the effectiveness of the method has been observed depending on whether the captured traffic on the host or the entire network is analyzed. Specifically, the EPS-based method has proven to be very effective over the traffic captured on the host, since the average packet size generated by legitimate users was 1275.77 bytes (Figure 5.3), which is logical since the vast majority of host traffic is user-data, unlike network capture traffic that contains many more communication-protocol packets that are considerably smaller in size. In contrast to the size of malicious packets, which have a fixed size of 540 bytes, the existence of only 3% of malicious users on the network already triggers the alarm signal.

On the other hand, for traffic captured on the network, the EPS method has not proved to be effective. This is because the average size of legitimate packets, 797.26 bytes (Figure 5.4), is closer to the size of malicious packets, 540 bytes. Despite following a clear pattern, attack traffic is not always perfect, as there are packets that are lost during transmission and therefore the standard deviation calculated on malicious traffic, which ideally should be 0, is modified. In the hypothetical case that the transmission of malicious packets was perfect and the standard deviation was 0, the EPS-based method should be just as effective for network-captured traffic as it has been shown to be for traffic captured directly on the host.

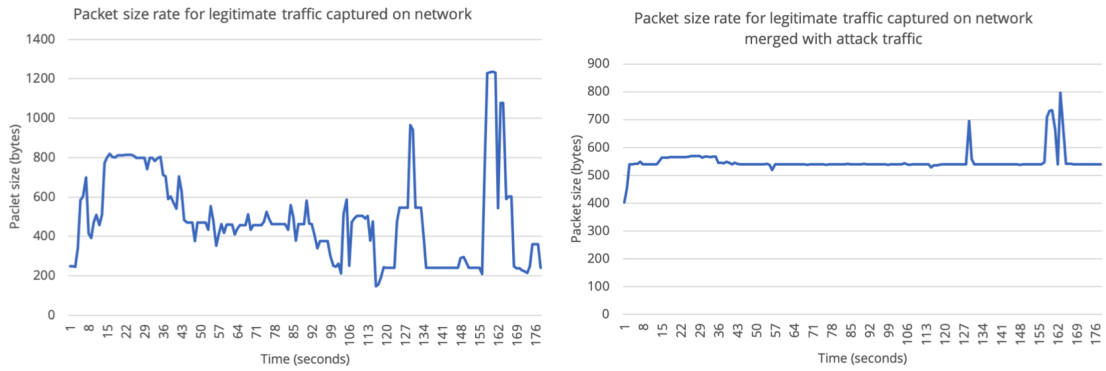


Figure 5.10: Comparison of packet size rate in a network-based source scenario for legitimate traffic and combination of legitimate and attack.

The relevance of using the standard deviation s to detect a relevant change in the behaviour of the analysed traffic can be seen in Figure 5.10. Without the existence of a series of attackers in the network that would produce a constant sending of packets of exactly the same size, the standard deviation from the size of the packets in that network is a considerably high value. However, by adding all these malicious users to the traffic, we observe how the average packet size per second is considerably normalized. In this way, the packets sent by the attackers produce a change in the behaviour of the traffic which, although it would not be detectable due to the fact that the size of these packets is between the thresholds considered for legitimate traffic, the calculation of the standard deviation allows us to detect it.

Nevertheless, the reason why EPS detection is more effective on traffic directly captured on the host is because the difference between the average EPS of the traffic under attack and the legitimate traffic is much smaller than in the capture scenario over the entire network. If the expected average packet size is very similar between the two types of traffic, the standard deviation of the one under attack must be very small for detection to be effective. This is why it has been determined that the effectiveness of the EPS method should be primarily implemented for detection directly at the victim host.

Finally, Figure 5.11 shows the successful detection ratio for the two methods presented, based on the number of attackers present, depending on whether the traffic was captured on the host or on the entire network. The number indicated in each cell represents the distance gap calculated according to Formulas 4.5 and 4.11 for the method based on entropy and EPS respectively.

Detection effectiveness matrix for host-based source

# Attackers	ϕ-entropy method	EPS method
4	24,84	277,82
5	26,79	288,97
6	28,77	298,31
7	30,78	306,53
10	36,99	327,18
20	59,43	375,75

Detection effectiveness matrix for network-based source

# Attackers	ϕ-entropy method	EPS method
4	62,68	179,41
5	65,37	172,58
6	68,08	167,3
7	70,81	162,91
10	79,09	152,67
20	107,79	129,18

Figure 5.11: Detection effectiveness matrix for different number of attackers in host-based and network-based captures for traffic under attack.

Chapter 6

Conclusions and future work

6.1 Conclusions

From the work developed, both from the comparative analysis based on the results published by Behal et al. [21] and Zhou et al. [30] and from the experimental analysis carried out for the present work, the following conclusions have been obtained:

- From the results published by the researchers of both papers, it has been concluded that the EPS-based has a higher detection rate as well as a lower false positive rate.
- Since the experiments carried out by the authors of both proposed methods were performed by comparing legitimate traffic and attack traffic exclusively, the full effectiveness of the methods developed cannot be concluded since it has not been considered that a real DDoS attack is conducted with a combination of attack traffic and legitimate traffic.
- The entropy-based detection metric has demonstrated to be effective if at least the 22% of the users are malicious when monitoring network-based traffic. For host-based traffic monitoring, the minimum percentage of attackers should be 65% in order to effectively detect the occurrence of a low-rate DoS attack.
- It has been experimentally demonstrated that the method of detection based on expectation of packet size is more effective when it is carried out by monitoring traffic directly on the victim host than if all the network traffic where the victim is located is analysed.

6.2 Future work

In order to generate malicious traffic in a controlled environment that is similar in its characteristics to real Internet traffic, it would be necessary to develop a network simulator instance –based, for example, on the ns-3 simulator– that would allow this traffic to be generated as close as possible to the expected behaviour. However, the development

of this task is beyond the scope of this work, but it is especially relevant to confirm the effectiveness of the methods analyzed applied to much larger networks.

In addition, traffic could be generated in real time, which would allow a process of adjustment of the properties of traffic considered legitimate, so that these properties are adjusted in real time to events that could affect the overall performance of the network without these being attacks, but simply a massive increase in users connected to the network under study, which is traditionally known as flash events.

Finally, to increase the speed of detection especially in large networks where the number of users and sending packets is much higher than in a home network, it would be very interesting to implement the methods studied in an FPGA board. The initial intention of this work consisted in the implementation of a DoS attack detection system in an FPGA. However, during the investigation of the different possible classifications of DoS attacks and due to the fact that there are already numerous effective detection methods for those considered high-rate, it was decided to study some existing detection methods of low-rate attacks for which such an exhaustive development had not been carried out by the scientific community.

Conclusiones y trabajo futuro

Conclusiones

A partir del trabajo desarrollado, tanto del análisis comparativo basado en los resultados publicados por Behal et al. [21] y Zhou et al. [30] como del análisis experimental realizado para el presente trabajo, se han obtenido las siguientes conclusiones:

- De los resultados publicados por los investigadores de ambos trabajos, se ha concluido que el método basado en EPS tiene una mayor tasa de detección, así como una menor tasa de falsos positivos.
- Como los experimentos realizados por los autores de ambos métodos propuestos se realizaron comparando tráfico legítimo y tráfico de ataque exclusivamente, no se puede concluir la plena eficacia de los métodos desarrollados, ya que no se ha considerado que un ataque DDoS real se realiza con una combinación de ambos tipos de tráfico.
- La métrica de detección basada en la entropía ha demostrado ser efectiva si al menos el 22% de los usuarios son maliciosos mientras se monitoriza el tráfico capturado en la red. Para la monitorización del tráfico directamente en la máquina víctima, el porcentaje mínimo de atacantes debería ser del 65% para detectar eficazmente la ocurrencia de un ataque DoS low-rate.
- Se ha demostrado experimentalmente que el método de detección basado en el tamaño de paquete esperado es más eficaz cuando se lleva a cabo monitorizando el tráfico directamente en el host atacado que si se analiza todo el tráfico de la red a la que se encuentra conectada la víctima.

Trabajo a futuro

Para generar tráfico malicioso en un entorno controlado y de características similares al tráfico real de Internet, sería necesario desarrollar un simulador de red (basado, por ejemplo, en el simulador ns-3) que permita generar este tráfico lo más similar posible al comportamiento esperado. Sin embargo, el desarrollo de esta tarea está fuera del alcance de este trabajo, pero resulta especialmente relevante para confirmar la efectividad de los métodos analizados aplicados a redes de mucho mayor tamaño.

Además, se podría generar tráfico en tiempo real, lo que permitiría un proceso de ajuste de las propiedades de tráfico consideradas legítimas, de forma que dichas propiedades se ajusten en tiempo real a eventos que puedan afectar al rendimiento global de la red sin que éstos sean ataques, sino simplemente un aumento masivo de usuarios conectados a la red bajo estudio, lo que tradicionalmente se conoce como eventos flash.

Finalmente, para aumentar la velocidad de detección especialmente en redes extensas donde el número de usuarios y envío de paquetes es mucho mayor que en una red doméstica, sería muy interesante implementar los métodos estudiados en una placa FPGA. La intención inicial de este trabajo consistió en la implementación de un sistema de detección de ataques DoS en una FPGA. Sin embargo, durante la investigación de las diferentes clasificaciones posibles de ataques DoS y debido a que ya existen numerosos métodos de detección eficaces para los considerados ataques high-rate, se decidió estudiar algunos métodos existentes de detección de ataques low-rate para los que la comunidad científica no había llevado a cabo un desarrollo tan exhaustivo.

Bibliography

[1] Alefiya Hussain, John Heidemann, and Christos Papadopoulos. A Framework for Classifying Denial of Service Attacks-Extended. Technical Report ISI-TR-2003-569b, USC/Information Sciences Institute, 06 2003. (Original TR, February 2003, updated June 2003).

[2] Qiao Yan, Fei Richard Yu, Qingxiang Gong, and Jianqiang Li. Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges. *IEEE Communications Surveys & Tutorials*, 18:602–622, 2016.

[3] Gaurav Somani, Manoj Singh Gaur, Dheeraj Sanghi, Mauro Conti, and Rajkumar Buyya. DDoS attacks in cloud computing: Issues, taxonomy, and future directions. *Computer Communications*, 107:30–48, 7 2015.

[4] Steve Mansfield-Devine. The growth and evolution of DDoS. *Network Security*, 2015(10):13– 20, 2015.

[5] Charalampos Patrikakis, Michalis Masikos, and Olga Zouraraki. Distributed Denial of Service Attacks. *The Internet Protocol Journal*, 7(4):13–35, 12 2004.

[6] Shui Yu. *Distributed Denial of Service Attack and Defense*. Springer Publishing Company, Incorporated, 2013.

[7] Nazrul Hoque, Hirak Jyoti Kashyap, and Dhruba Kumar Bhattacharyya. Real-time DDoS attack detection using FPGA. *Computer Communications*, 110:48–58, 2017.

[8] Steve Mansfield-Devine. The evolution of ddos. *Computer Fraud & Security*, 2014(10):15–20, 2014.

[9] Aftab Afzal. 25 years of DDoS. *Database and Network Journal*, 46(4), 08 2016.

[10] Ponemon Institute LLC. Cost of Cybercrime Study. Annual report of the ponemon institute llc jointly developed by accenture, Ponemon Institute LLC, 2019.

[11] Jose Nazario. DDoS Attack Evolution. *Network Security*, 2008(7):7–10, 7 2008.

[12] T. M. Wu. Intrusion Detection Systems. *Information Assurance Technology Analysis Center (IATAC)*, 09 2009.

[13] Mohammed Alenezi and Martin Reed. Methodologies for detecting DoS/DDoS attacks against network servers. *ICSNC 2012, The Seventh International Conference on Systems and Networks Communications*, pages 92–98, 11 2012.

[14] Sunny Behal and Krishan Kumar. Detection of DDoS Attacks and Flash Events Using Information Theory metrics - An Empirical Investigation. *Computer Communications*, 103(C):18–28, 05 2017.

[15] Yang Xiang, Ke Li, and Wanlei Zhou. Low-Rate DDoS Attacks Detection and Traceback by Using New Information Metrics. *IEEE Transactions on Information Forensics and Security*, 6(2):426–437, 06 2011.

[16] Pedro García-Teodoro, Jesús E. Díaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1):18–28, 2009.

[17] Rudolf B. Blazek, Hongjoong Kim, Boris Rozovskii, and Alexander G. Tartakovsky. A novel approach to detection of "denial–of–service" attacks via adaptive sequential and batch–sequential change–point detection methods. 2001.

[18] Wei Lu and Ali A. Ghorbani. Network anomaly detection based on wavelet analysis. *EURASIP J. Adv. Signal Process*, 2009:4:1–4:16, January 2009.

[19] Lan Li and Gyungho Lee. Ddos attack detection and wavelets. *Telecommunication Systems*, 28(3):435–451, 03 2005.

[20] Jelena Mirkovic and Peter Reiher. A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. *ACM SIGCOMM Computer Communication Review*, 34:39–53, 2004.

[21] Sunny Behal and Krishan Kumar. Detection of DDoS Attacks and Flash Events Using Novel Information Theory Metrics. *Computer Networks*, 116(C):96–110, 04 2017.

[22] A. Kuzmanovic and E. W. Knightly. Low-rate tcp-targeted denial of service attacks and counter strategies. *IEEE/ACM Transactions on Networking*, 14(4):683–696, 08 2006.

[23] Changwang Zhang, Zhiping Cai, Weifeng Chen, Xiapu Luo, and Jianping Yin. Flow level detection and filtering of low-rate ddos. *Computer Networks*, 56(15):3417–3431, 2012.

[24] Paul Aitken, Benoit Claise, and Brian Trammell. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. RFC 7011, 9 2013.

[25] Rick Hofstede, Pavel Čeleda, Brian Trammell, Idilio Drago, Ramin Sadre, Anna Sperotto, and Aiko Pras. Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX. *IEEE Communications Surveys Tutorials*, 16(4):2037–2064, 2014.

[26] J Luo, Xiaolong Yang, Jin Wang, Jie Xu, Jian Sun, and Keping Long. On a Mathematical Model for Low-Rate Shrew DDoS. *IEEE Transactions on Information Forensics and Security*, 9:1069–1083, 07 2014.

[27] Monowar H. Bhuyan, D.K. Bhattacharyya, and J.K. Kalita. An empirical evaluation of information metrics for low-rate and high-rate ddos attack detection. *Pattern Recognition Letters*, 51:1–7, 2015.

[28] Robert M. Gray. *Entropy and Information Theory*. Springer-Verlag, Berlin, Heidelberg, 1990.

[29] P.K. Bhatia and Surender Singh. On a new Csiszar's f-divergence measure. *Cybernetics and information technologies*, 13(2):43–57, 2013.

[30] Lu Zhou, Mingchao Liao, Cao Yuan, and Zhang Haoyu. Low-Rate DDoS Attack Detection Using Expectation of Packet Size. *Security and Communication Networks*, 2017:1–14, 10 2017.

[31] Monika Sachdeva, Krishan Saluja, and Gurvinder Singh. A comprehensive approach to discriminate DDoS attacks from flash events. *Journal of Information Security and Applications*, 26, 12 2015.

[32] Center for Applied Internet Data Analysis. The CAIDA UCSD "DDoS Attack 2007" Dataset, 2007. `http://www.caida.org/data/passive/ddos-20070804_dataset.xml`.

[33] Center for Applied Internet Data Analysis. CAIDA UCSD Network Telescope Traffic Dataset, 2001-2008. `http://www.caida.org/data/passive/backscatter_dataset.xml`.

[34] WIDE-TRANSIT 100 Megabit Ethernet Trace 2007-01-09, 2007. `http://imdc.datcat.org/collection/1-055M-0=WIDE-TRANSIT+100+Megabit+Ethernet+Trace+2007-01-09+`.

[35] MIT Lincoln Laboratory. LLSDDOS0.2.2 Dataset, 2000. `https://www.ll.mit.edu/ideval/data/2000data.html`.

[36] FIFA World-cup 1998 Dataset, 1998. `http://ita.ee.lbl.gov/html/contrib/worldcup.html`.

[37] Balakrishnan Chandrasekaran. Survey of network traffic models. *Washington University in St. Louis CSE*, 567, 2009.

[38] Dharanikota, Samvid and Bharadwaj, Sagar and Honawad, Adarsh. Low-Rate-TCP-DoS-Attack. `https://github.com/samvid25/Low-Rate-TCP-DoS-Attack`, 2017.

[39] ns-3 Network Simulator. `https://www.nsnam.org/`.

[40] Wireshark Network Protocol Analyzer. `https://www.wireshark.org/`.

[41] SiLK – CERT NetSA Security Suite. https://tools.netsa.cert.org/silk/.

[42] Python Programming Language. https://www.python.org/.